

# Final Project Report

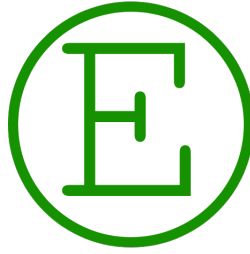
## Course Evaluation System

Team EVAL

Jovon Craig, Sam Elliott, Yuanqi Guo, Robert Judkins, and Stanley Small

Client: Dr. Harlan Onsrud

May 7, 2019



# Course Evaluation System

## Final Project Report

### **Contents**

# 1 Introduction

## 1.1 Purpose of This Project

For our capstone project, Team EVAL created the Course Evaluation System, a web application designed to be intuitive and versatile for college instructors. This new system, which will be tested first at the University of Maine, allows one to quickly create course evaluation surveys and review their responses. We undertook this project for our client, Dr. Harlan Onsrud, who wanted a more efficient way to process course evaluations. By making and documenting the software, the team has gained valuable skills in software engineering, programming, and technical writing.

## 1.2 Purpose of This Document

This final project document gives an overview of our course evaluation system, the purpose of it, and why we believe it is useful. We first talk about existing evaluation creation software and how it is not tailored to college instructors and administrators. The next section goes into more detail about the system, discussing its requirements, architecture, development process, and user interface. The last section describes how the finished product improves productivity, our thoughts on what we learned from the project, and the steps we would take to improve the product.

This document is intended for the development team, the product client, and potential users of the system. Team EVAL needs this document to summarize the product's development in job interviews. The client, Professor Harlan Onsrud, also needs it to verify that we designed the system according to his needs. The document helps the software's users by informing them what the system does and how it particularly helps them with their work.

## 1.3 References

Craig, J., Elliott, S., Judkins, R., & Small, S. 29 October 2018. *System Requirements Specification*.

Craig, J., Elliott, S., Judkins, R., & Small, S. 16 November 2018. *System Design Document*.

Craig, J., Elliott, S., Judkins, R., & Small, S. 30 November 2018. *User Interface Design Document*.

Craig, J., Elliott, S., Guo, Y., Judkins, R., & Small, S. 14 March 2019. *Code Inspection Report*.

Craig, J., Elliott, S., Guo, Y., Judkins, R., & Small, S. 4 April 2019. *Administrator Manual*.

Craig, J., Elliott, S., Guo, Y., Judkins, R., & Small, S. 18 April 2019. *User Manual*.

Onsrud, H. "Example Question Selection Form." See Appendix H.

Onsrud, H. "Example Results Display." See Appendix I.

LimeSurvey: The online survey tool - open source surveys. (n.d.). Retrieved from <https://www.limesurvey.org/>.

# 2 Purpose of This System

## 2.1 Our Problem

In 2015, there were more than 4,600 higher-education institutions in the United States. Virtually all of them offer a breadth of college courses for students to complete and carry them through their academic careers. At the end of a course, students typically fill out an evaluation manually so that teachers know how they are performing in their roles. Unfortunately, many institutions, such as the University of Maine, are behind in the technology that can be used for this task.

The University of Maine has traditionally used Scantron sheets for course evaluation forms. As Team EVAL knows, filling in tiny bubbles with pencil and paper is tedious. It is also hard work for the college

administration to use Scantron sheets. They need to scan the forms for every student in every course, and the data then needs to be collected as forms that are readable for instructors. The University of Maine must upgrade its process to keep with the times and exploit today's technology.

## **2.2 Existing Survey Software**

Several software solutions currently exist that allow users to create, modify, and publish surveys. One notable example is LimeSurvey, a free and open-source survey creation tool. The team acknowledges the immense amount of work put into creating LimeSurvey, with its selection of question types, scalability, visualization features, and other powerful functionality. However, LimeSurvey has always been general-purpose software; it is aimed at multiple types of users who want to make surveys. Consequently, it is complex and has a high learning curve.

College teachers and administrators need to jump through hoops to use tools like LimeSurvey, which are not always easy or intuitive. Their evaluations are often highly standardized and sent to multiple different classes, yet LimeSurvey and similar software do not account for these qualities. A college instructor would have to manually input every field, question, and class roll into each evaluation from scratch. UMaine is using another service, Blue by Explorance, that has the necessary functionality, but it is expensive to use for a university and often inflexible if administrators want to differ from the norm. These are some of the reasons why our client, Dr. Onsrud, wants us to create software that is better suited to instructors. We believe our Course Evaluation System will help instructors make their own surveys efficiently and easily.

## 3 About This System

### 3.1 System Requirements

Near the beginning of the project, we met with Professor Onsrud to solicit the system requirements, what the evaluation system must do to function as Dr. Onsrud intends. The System Requirements Specification (SRS) divides the requirements into two categories. The functional requirements specify the major actions a user can perform with the system. They are listed in Table 1 below:

Number	Priority	Name	Description
1	5	Log in to system	A user logs in with a Google e-mail address and password
2	5	Create evaluation	A user creates a new evaluation form for a class
3	5	Edit evaluation	A user enters and edits information and questions for an evaluation form
4	5	Publish evaluation	A user publishes an evaluation form
5	5	View evaluation results	A user reviews the survey results of a category of courses

Table 1: Functional requirements

The non-functional requirements state the system’s qualities that do not pertain to how the program behaves. They are listed below in Table 2:

Number	Priority	Description
1	3	The software should be supported by the latest versions of Windows, Mac, Linux, iOS, and Android.
2	4	The software should be accessible by the latest versions of Safari, Chrome, Firefox, and Edge.
3	5	All questions entered by the teacher or administrator shall appear on the output survey.
4	5	All data stored in the program’s database shall be valid.
5	5	All collected survey data shall not be alterable.
6	4	Teachers shall not be able to access data of courses other than their own.
7	3	The mean time between failures should be at least 60 minutes.
8	5	Students shall have no access to any data stored by the program.
9	5	All survey responses (except signed comments) shall be anonymous.
10	2	The software should scale to at least three universities, 1000 courses per semester, 1000 teachers per university, and 500 students per course.
11	1	The software should not exceed 500 MB in size.
12	4	The software’s source code shall be open-source and shall use a GPLv2 license.
13	4	The licensing requirements of any non-original code shall be met.
14	4	The software shall meet UMaine AFUM requirements.

Table 2: Non-functional requirements

## 3.2 System Architecture

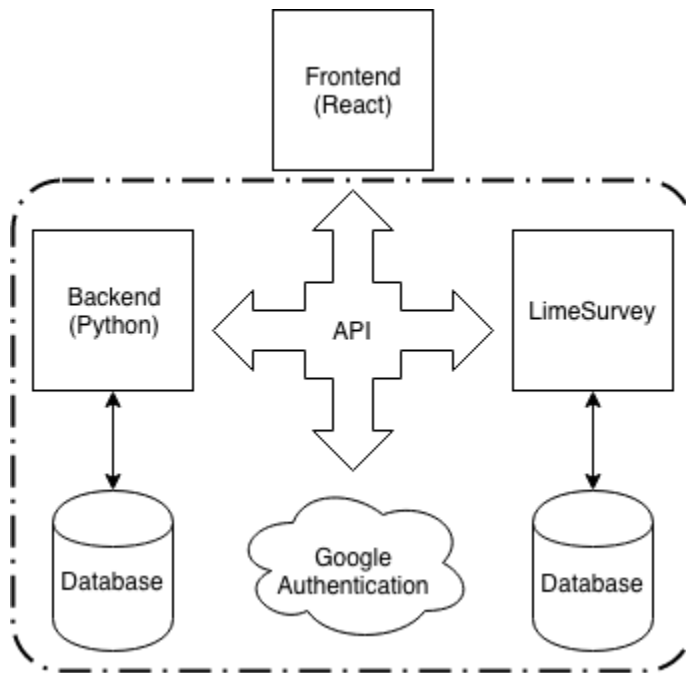


Figure 1: Component diagram of the system

Our system features a modular design, with many distinct parts that communicate via an API. The front end implements the user interface, and it is written in JavaScript and the React framework. The back end communicates with the system’s databases and is written in Python and the Flask framework. We are using two databases, one maintained by LimeSurvey to store evaluation responses, and another maintained by our system to store question data. Additionally, the API will communicate with Google Authentication to authenticate users. The design and implementation of our system can be found in the System Design Document (SDD). The system’s components are displayed in Figure 1 above.

By maintaining a modular approach to the design of the system, team members can work independently on separate components of the system. Moreover, by keeping the LimeSurvey codebase separate from our system, we can update our original code without disrupting the functionality of LimeSurvey.

## 3.3 Our Process

As our project required following a strict timeline over the 2018-2019 academic year, Team EVAL used a software development process that subscribed to both waterfall and agile methodologies. We primarily used the Scrum framework to develop our product. The team held stand-up meetings twice a week, as well as biweekly Scrum planning meetings. We also followed the V-model, a waterfall-like process that emphasizes verification in the first half of development and validation in the second half. A diagram of the V-model is shown as Figure 2 on the next page.

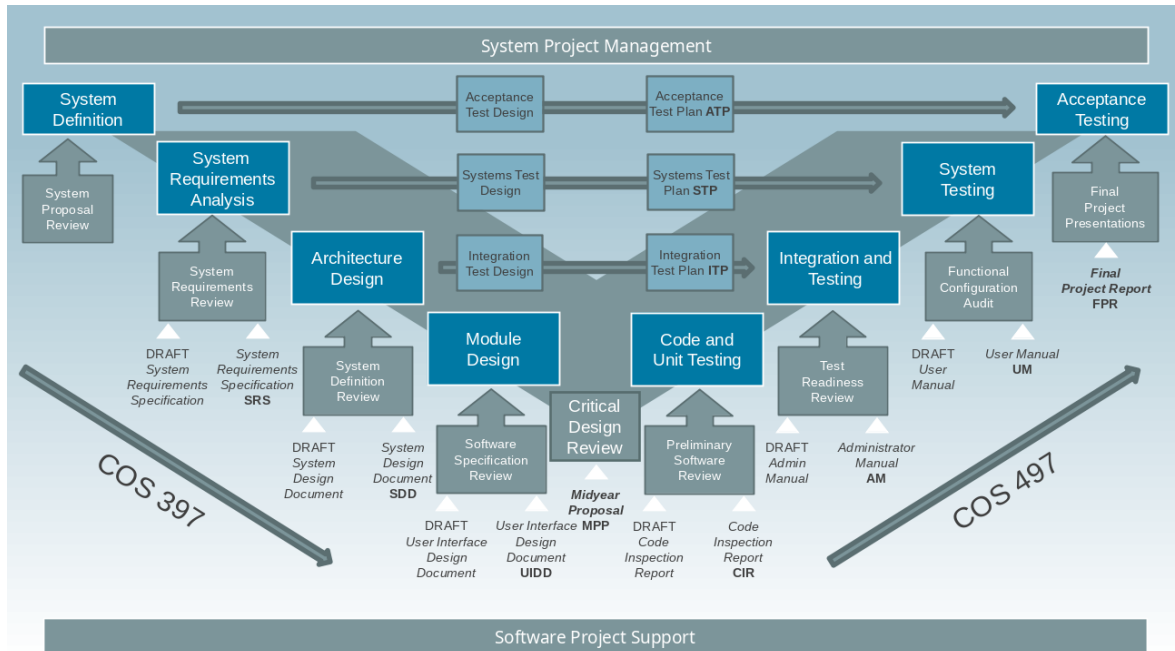


Figure 2: The V-model, showing the steps in the process

The team took additional measures to organize our development of the system. We assigned team roles according to the Team Software Process: the team leader, the development manager, the planning manager, the process manager, and the support manager. In addition, the team frequently collaborated with the client to ensure the correct product was built. As for tools, GitHub was used to store and maintain the code, and Docker allowed our code could run in a familiar environment on different machines.

Team EVAL wrote numerous tests over the course of the project to ensure that the course evaluation system met its functional requirements. On the back end, Jovon and Tom wrote several unit tests for each endpoint, both positive and negative, with Python’s “unittest” library. These tests check whether the routes can handle API calls with a variety of input. Sam wrote additional tests for the front end using WebdriverIO. He included unit tests that validate functions specific to the user interface, such as loading data into text fields, and integration tests that interact with the API to validate whether the UI is properly using our database.

### 3.4 User Interface

The user interface is a pivotal part of the user experience and is important for our product. Our goal of the system was to make an easy-to-use software that allows users to create course evaluations for their classes in a web application. Users have no interaction with the back-end architecture and can only see the user interface, so the ease of use relies entirely on the front-end UI. Detailed descriptions of every aspect of our user interface can be found in the User Interface Design Document (UIDD).

The overarching theme for the user interface is "less is more". The interface has as few extraneous elements as possible, is as intuitive as possible, and does not require a large amount of effort to understand the system. To this end, all functionality of the product is accessible from the home page, of which a screenshot is shown below as Figure 3.

**Wicked Easy Teaching Evaluations**

HomeAboutFAQLogout

---

**1. Create a New Course Evaluation Form**

To create a new course evaluation form click the **Create** button bellow. You will be able to copy a template from previous evaluations you have worked on or start from a blank evaluation.

Create

**2. Edit an Existing Unpublished Course Evaluation Form**

To edit a course evaluation form that you have previously created but not published select it from the drop down and click on the **Edit** button bellow. You will be given the opportunity to published the evaluation upon saving any changes made.

Select an evaluation ▾Edit

**3. View Evaluation Results**

To view results of your evaluations select which type of report you would like to view and select from the drop down which report you want to view then click the **View Results** button.

☒ Instructor☐ Course Designator☐ Unit☐ College☐ University

Select an Instructor ▾View Results

Figure 3: Home screen of the user interface

After passing the landing screen, the user is instantly aware of the functions that one can make through the home screen. The interface allows users to create new evaluations, view their existing and previous evaluations, and view statistics for the results of completed evaluations. Users can also select an existing form as a preset for a new evaluation. To reduce confusion, help text for different features is present on the same page and does not require users to navigate away from it.



Our user interface was designed to emulate the physical processes that have been in place at the University of Maine for administering teaching evaluations. This decision was made to make the transition to our software easier for all parties involved. Our software collects the same information as earlier systems did.

**Wicked Easy Teaching Evaluations** Home About FAQ Logout

If you want information on this form auto-filled from a previous submission that you may then edit, please select a past evaluation to copy:

Select an evaluation Select

Next

Please enter information pertaining to the course you are going to be evaluating. Often times the Course Evaluation Administrator and the name of the Instructor are the same.

**Course Information**

Course Designator (e.g. MUS) *	
Course Number (e.g. 200) *	
Course Section (e.g. 001) *	
Course Title (e.g. Ballroom Dance) *	
Is this a graduate course? (Y/N) *	
Semester and Calendar Year (e.g Fall 2019) *	
Name of Faculty Unit ( e.g. School of Music) *	
Name of College ( e.g. Liberal Arts) *	

Figure 4: The first screen seen when creating an evaluation

Once a user decides to create an evaluation, they are met with a simple screen with several forms for them to fill out, as shown in Figure 4 above. The interface is designed to be self-explanatory so that anyone can create an evaluation with no troubles. The user is presented with three forms to fill out the information about their course, their students, and the questions they would like to include in their evaluation.

Overall, our interface is designed with the hopes that any user could use our product easily and without any hold-ups. They can effortlessly go through the various screens to create, edit, or publish an evaluation and view the results once their students have completed the evaluation.

## 4 Conclusion

### 4.1 Needs This System Meets

Our course evaluation system will bring the University of Maine into the modern era of using technology to evaluate courses. Using our software, an instructor can choose questions for their surveys with ease, and students are able to take the surveys anytime and anywhere. It is convenient to publish a survey, as LimeSurvey handles its creation, distribution, and e-mails to its participants. The results are automatically fed back into the software, and the instructor can view statistics of the results and download them afterwards. The administrator no longer needs to manually distribute evaluations, collect them, and scan them, and then do statistical analysis to get their results. The system does these functions automatically, making the evaluation process significantly easier for administrators and teachers.

Furthermore, this product eliminates the need for Scantron sheets, saving paper and the need for students to dedicate time in class to fill out bubble sheets. It instead provides a fast and simple way to distribute evaluations to all students in a desired course. Lastly, the product meets Dr. Onsrud's most important requirements. It is free and open-source, and it is compatible with all major web browsers. All survey responses are anonymous, and the data stored by the program cannot be accessed by students. Eventually, our system will be free to use for instructors at the University of Maine and by any teachers elsewhere.

### 4.2 What We Have Learned

Throughout the process of developing the system, the team learned numerous lessons. Members of the team will attempt to incorporate the values of these lessons into future projects and our professions.

One of the first lessons learned in the project was the importance of active and effective communication. Any progress made by the team relies on communication and teamwork. Although the team did use Slack to communicate, we should have added the client into our Slack channel to keep the client informed and more active in the project. As the client spent more time away from the project in the second semester, productivity and progress waned.

Another lesson learned was that defining and following a process is important for developing a working and effective product. Without adhering to the process defined by Dr. Yoo, progress would have been greatly diminished. The process allowed the team to ensure that we stayed on task and continued working throughout the course of the project.

The team also realized that early and constant feedback is essential, especially from the client. As mentioned previously, the inability of the team to effectively communicate with the client during the second semester of the project made it difficult to complete certain functionality to the needs of the client. While process remains important, the ability to always have a working implementation remains key. The team should have worked on creating a minimum viable product first, then receiving feedback later.

Finally, the team learned that software building is not equivalent to coding. As Fred Brooks states, only about one sixth of a team's time should be spent coding. The rest of the time is spent planning and testing, to ensure the team creates the right product. Verification and validation are key components of building software that both the team and the client are happy with. Throughout the process, the team learned that system development depends on much more than just coding.

### 4.3 Future Work

Despite working on the project for an academic year with five team members, Team EVAL did not achieve everything we set out to accomplish. This section describes the elements and features the team would have liked to include in the system, had time allowed.

First, the team wishes to improve the appearance and functionality of the user interface. The current version of the project represents the first generation of what could potentially be a very user-friendly product. The team wishes to have groups for testing and evaluating the evaluation system to ensure that it accomplishes all of the goals defined by the client.

Second, the team would like to expand on results reporting. The results accessible to the system's users remains limited at this point in time. The team wishes to expand the reporting capabilities of the system to

provide more advanced statistics, representing feedback for instructors, the college, and the schools within the system.

Additionally, the team would like make the evaluation structure more flexible. Currently, the system is tuned to the needs of the University of Maine for teaching evaluations. To make the system more usable and accessible to a wider audience, the team would like to provide the opportunity for more customization and personalization of evaluations to system users.

Finally, we wish to automate the deletion of data after the semester concludes. As anonymity and security remain some of the key components of the teaching evaluation system, the team would like to discard information that is not currently being used.

## 5 Acknowledgments

We would like to thank multiple people for giving us the opportunity to work on the Course Evaluation System. We thank Dr. Harlan Onsrud for letting us work on the course evaluation system and being patient with us during the development process. We thank Dr. Terry Yoo for guiding us in our capstone project and teaching us good software engineering practices. We could not have finished the project without Yoo. Finally, we thank the developers of LimeSurvey for providing an open-source survey creation program that we could incorporate into our software, saving us development time.

## A Document Contributions

This section lists the contributions that each member of Team EVAL made to this document.

Stanley Small wrote the system architecture section and most of the conclusion, including the “What We Have Learned” and “Future Work” sections. He also added Figure 4 to the document.

Jovon Craig wrote the system requirements section, “Needs This System Meets” section, and most of the introduction, “Purpose of This System” section, and process section. He also made revisions to the whole document.

Sam Elliott wrote most of the user interface section and contributed to the testing section. He made some revisions to the document as a whole.

Robert Judkins wrote a part of the user interface section and made a few revisions to the conclusion.

Yuanqi Guo added more content to the “Existing Survey Software” and “Needs This System Meets” sections.

## **B System Requirements Specification**

### System Requirements Specification

Team EVAL

October 29, 2018



# College Course Evaluation System

## System Requirements Specification

### Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose of this Document . . . . .	3
1.2	References . . . . .	3
1.3	Purpose of the Product . . . . .	3
1.4	Product Scope . . . . .	3
<b>2</b>	<b>Functional Requirements</b>	<b>6</b>
2.1	Tests . . . . .	10
<b>3</b>	<b>Non-Functional Requirements</b>	<b>11</b>
3.1	Tests . . . . .	13
<b>4</b>	<b>User Interface</b>	<b>14</b>
<b>5</b>	<b>Deliverables</b>	<b>14</b>
<b>6</b>	<b>Open Issues</b>	<b>14</b>
<b>A</b>	<b>Agreement Between Customer and Contractor</b>	<b>15</b>
<b>B</b>	<b>Team Review Sign-off</b>	<b>16</b>
<b>C</b>	<b>Document Contributions</b>	<b>17</b>

# 1 Introduction

## 1.1 Purpose of this Document

This system requirements specification details what our course evaluation system does and what tests we will make to ensure the system is complete. It includes why we are creating the system, the scope of the product, diagrams that illustrate the system, what we will deliver to the customer, and currently pending issues about the software.

This document is intended for the development team, the product client, Dr. Harlan Onsrud, and potential users of the system. Team EVAL needs this document to ensure that the product works as intended. Dr. Onsrud needs it to know that he will receive the program that he desires. The document also helps the software's users in that they can learn more about the functions of the evaluation system.

## 1.2 References

LimeSurvey: The online survey tool - open source surveys. (n.d.). Retrieved from <https://www.limesurvey.org/>

Fowler, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Boston: Addison-Wesley.

## 1.3 Purpose of the Product

The University of Maine gives out course evaluation surveys to students at the end of each course. The survey is filled on a bubble sheet and is then scanned. Harlan Onsrud finds it inconvenient for the school administrators to manually scan and compile the survey results. Current campus experiments with electronic evaluations are not fully automated and do not seem to fulfill the desires of students and faculty. He desires an online, automated evaluation system that is responsive to the University's community needs and improves productivity.

## 1.4 Product Scope

Team EVAL will create a product which interfaces with an already existing survey software, LimeSurvey, to provide both individual teachers and administrators the ability to create and administer evaluations. This product will be usable even by those without technical backgrounds, and it will have an intuitive interface for setting up and administering teacher evaluations.

It will allow users to create one or more courses with predefined survey data and rosters of e-mail addresses. For each class, users will be able to choose from provided questions or enter their own custom questions. The question sets that they make may be saved to their account and applied to future evaluations they create. Upon request of the user, the product will create a LimeSurvey with the supplied information. It then sends an invitation to complete the survey to the students on the class' roster via e-mail, along with reminders as appropriate. When the survey is terminated by the instructor/administrator or by a certain data and time, the product will allow users to view and download a statistical analysis and clear visualization of the data collected for one or more of their courses.

The software will support the reporting of accumulated data and the appropriate statistics and graphics derived from it for each academic period. A report will be generated for the following levels: course section, all sections of the same course, all courses of each instructor, all courses with the same designator (e.g. "COS"), all courses in the same department/school, all courses in the same college, and all courses in the same university.

The product will be completed in time to administer teacher evaluations for the spring of 2019 at the University of Maine, whether evaluation forms are created by instructors or administrators. A UML diagram (Figure 1) shows the scope as a dotted rectangle.

LimeSurvey is an existing software that has the ability to create and publish surveys. This product will simply interface with LimeSurvey and interact with its components. LimeSurvey lies outside of the scope of this project and will be treated as a third party actor.

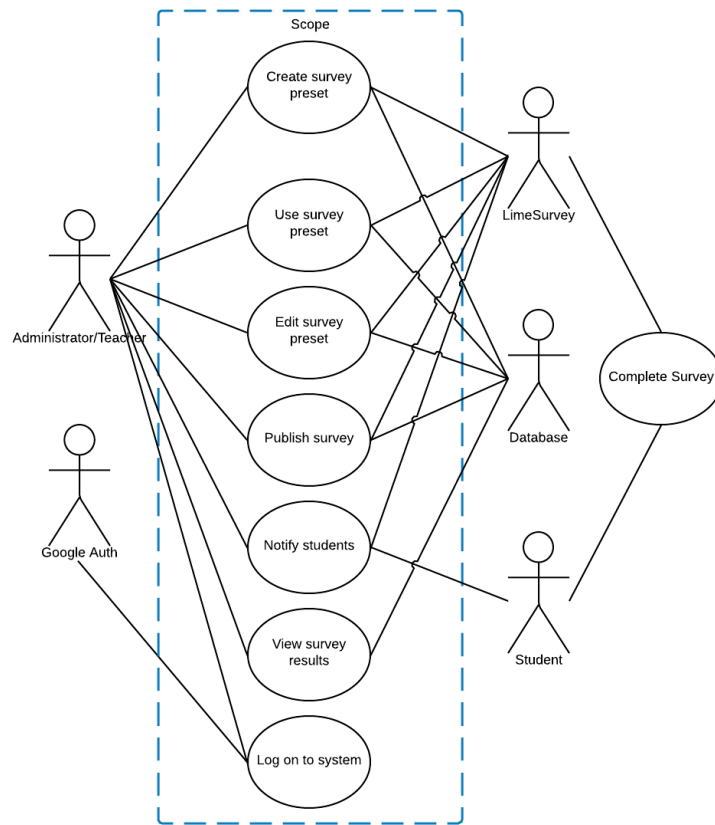


Figure 1: Use Case diagram of the system scope.





## 2 Functional Requirements

The functional requirements specify what actions the program will perform. Each requirement is represented as a use case. The UML diagram showing the use cases is on the next page.

Table 1

<b>Number</b>	1	
<b>Name</b>	Log on to system	
<b>Summary</b>	A teacher or administrator logs in with a user-name and password	
<b>Priority</b>	5	
<b>Preconditions</b>	The user has an Google account (or can otherwise sign in with Google)	
<b>Postconditions</b>	A session is started and an authentication token is sent to the program	
<b>Primary Actors</b>	Administrator, Teacher	
<b>Secondary Actors</b>	Google Authentication	
<b>Trigger</b>	A user attempts to log in	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Open the log-in page
	2	Enter Google log-in information
	3	User is authenticated
	4	User gains access to the system
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Incorrect log-in info : Do not proceed and alert user
<b>Open Issues</b>	What method to use for authentication of non-anonymous comments by students	

Table 2

<b>Number</b>	2	
<b>Name</b>	Create survey preset	
<b>Summary</b>	An administrator or teacher adds a preset list of questions to the database	
<b>Priority</b>	5	
<b>Preconditions</b>	A course's evaluation survey questions are unchosen or unwritten	
<b>Postconditions</b>	The course's evaluation survey is written	
<b>Primary Actors</b>	Administrator, Teacher	
<b>Secondary Actors</b>	Database, LimeSurvey	
<b>Trigger</b>	A new course needs to be evaluated	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Select option to add a preset
	2	Enter the survey data on LimeSurvey
	3	Database updated with survey info
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Invalid data is entered : Notify user of incorrect data
<b>Open Issues</b>	What information about the survey shall be stored, what format should the data be in	

Table 3

<b>Number</b>	3	
<b>Name</b>	Use survey preset	
<b>Summary</b>	An administrator or teacher retrieves a preset for a survey developed previously for a course	
<b>Priority</b>	4	
<b>Preconditions</b>	The survey preset exists in the database	
<b>Postconditions</b>	A preset list is available to create a survey	
<b>Primary Actors</b>	Administrator, Teacher	
<b>Secondary Actors</b>	Database, LimeSurvey	
<b>Trigger</b>	A user wants the same questions as on a previous survey	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Select option to use a preset list of questions
	2	Choose which preset to use
	3	Preset data from database converted to LimeSurvey form
	4	LimeSurvey updated with preset
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Data in preset is invalid : Notify user of invalid data
<b>Open Issues</b>	Which presets are visible to each teacher	

Table 4

<b>Number</b>	4	
<b>Name</b>	Edit survey preset	
<b>Summary</b>	An administrator or teacher edits the survey questions for a course	
<b>Priority</b>	5	
<b>Preconditions</b>	A course for the survey questions exists in the database	
<b>Postconditions</b>	A course includes appropriate survey data	
<b>Primary Actors</b>	Administrator, Teacher	
<b>Secondary Actors</b>	Database, LimeSurvey	
<b>Trigger</b>	A college course survey has incorrect data or preferences	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Choose the course whose questions are to be edited
	2	Select questions preset if applicable
	3	Modify the data to be appropriate for the course
	4	Database updated with new edits
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	3a	Invalid data is entered : Notify user of incorrect data
<b>Open Issues</b>	None	

Table 5

<b>Number</b>	5
<b>Name</b>	Publish survey
<b>Summary</b>	LimeSurvey or the evaluation software is used to send out a survey to students
<b>Priority</b>	5
<b>Preconditions</b>	The chosen class has questions entered
<b>Postconditions</b>	A survey is created using LimeSurvey and sent to the students on the class roster
<b>Primary Actors</b>	Administrator, Teacher
<b>Secondary Actors</b>	Database, LimeSurvey
<b>Trigger</b>	A teacher or administrator wants to send out an evaluation
<b>Main Scenario</b>	<b>Step</b> <b>Action</b>
	1      Select option to publish survey
	2      Select course(s)
	3      A survey is created using LimeSurvey
	4      The survey is sent to all students on the class roster
<b>Open Issues</b>	None

Table 6

<b>Number</b>	6
<b>Name</b>	Notify students
<b>Summary</b>	LimeSurvey or the evaluation software is used to e-mail students to complete the survey, giving up to 3 automatic reminders as appropriate
<b>Priority</b>	5
<b>Preconditions</b>	A course evaluation survey is completed
<b>Postconditions</b>	All students are finished with their survey, or a deadline is reached
<b>Primary Actors</b>	Teacher
<b>Secondary Actors</b>	Student, LimeSurvey
<b>Trigger</b>	The teacher releases the survey to students
<b>Main Scenario</b>	<b>Step</b> <b>Action</b>
	1      Send initial reminder to students about survey using course roster
	2      Students complete survey
	3      Copy LimeSurvey results to database
	4      Database deletes data that is 60 days old
<b>Extensions</b>	<b>Step</b> <b>Branching Action</b>
	2a      A student does not complete survey : Send another reminder to student
<b>Open Issues</b>	How often to remind students, how to handle signed comments

Table 7

<b>Number</b>	7	
<b>Name</b>	View survey results	
<b>Summary</b>	An administrator, teacher, or both receive an automated e-mail, which has a link to view or download the survey results (i.e. averages and charts)	
<b>Priority</b>	4	
<b>Preconditions</b>	Survey responses are entered in database	
<b>Postconditions</b>	All appropriate averages are computed and displayed	
<b>Primary Actors</b>	Administrator, Teacher	
<b>Secondary Actors</b>	Database	
<b>Trigger</b>	A user seeks information about the responses to a group of surveys	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Click link to view results
	2	Compute average score for each question
	3	Display results by instructor, department, and university
	4	Select option to download results
<b>Open Issues</b>	Which survey questions are averaged for display	

## 2.1 Tests

These are the tests that will verify the functional requirements:

1. (Use Case 1) The tester attempts to log on to the system using four different sets of credentials: a valid e-mail address and password of an existing account (with courses already created), a valid address and password of a nonexistent account, an address with an invalid password for an existing account, and an invalid e-mail address and password. The test is successful if the first set passes and the last three sets fail. In the second case, a user will be prompted to register for the system.
2. (Use Case 2) The tester selects the option to add a course preset, then submits evaluation questions, preferences, the course roll, and e-mail texts. The test is successful only if all the data entered is reflected in the database. The data should still be there after logging out of the system. The team will create several courses/sections of the same course to ensure that the data loaded into LimeSurvey remains unchanged from when the original course was loaded. We will test adding students' full names and e-mails to the list of enrolled students.
3. (Use Case 3) The tester creates a course survey. He or she then tries to select that same survey to be used again. The new preset should still be present on LimeSurvey. The test is successful only if the preset data is loaded correctly to LimeSurvey from the database. The tester will exit the evaluation system and log in again to ensure that the data is saved.
4. (Use Case 4) The tester selects the option to create a survey. Next, he or she chooses a questions preset for the survey. The tester then adds one question and edits an existing question from the preset. After submitting, the database should include the new survey, containing the preset list, added question, and edited question. The team will run the program several times, each time selecting different questions to be included in the survey. We will ensure that some questions are mandatory and must be included no matter what.
5. (Use Case 5) The tester creates a course survey, does not enter any questions, and attempts to send the survey. The tester then enters some questions and sends the survey again. The team verifies that the survey is created with all questions entered and that the e-mail addresses on the course roster receive a prompt to complete it.
6. (Use Case 6) The tester messages a group of ten unique people about a mock survey. In a successful test, all ten people receive an e-mail with each containing a link to the survey. Most respondents promptly complete the survey, but one does not. The database must contain the survey responses of the nine people who finished, and the program must send another reminder e-mail to the unfinished person about the survey. Finally, the database must delete old survey data. The team will set the time periods before the reminder e-mail and data deletion to be very short to test whether the two functions work.
7. (Use Case 7) The tester selects the option to view the survey results. There are survey results using mock data for (a) each course section, (b) all sections of the same course, (c) all courses of each instructor, (d) all courses with the same designator, (e) all courses in the same department/school, (f) all courses in the same college, and (g) all courses in the same university. The team will add several surveys, each with different questions. If the test is successful, the correct average scores for every question in the course evaluation survey are shown in the application. Bar graphs for each course and section are supplied as well. Also, the software must show the appropriate results for all seven levels listed above.

### 3 Non-Functional Requirements

The non-functional requirements state the qualities of the program that are unrelated to its function.

Table 8

<b>Number</b>	1
<b>Priority</b>	3
<b>Description</b>	The software should be supported by the latest versions of Windows, Mac, Linux, iOS, and Android.
<b>Test Number(s)</b>	1

Table 9

<b>Number</b>	2
<b>Priority</b>	4
<b>Description</b>	The software should be accessible by the latest versions of Safari, Chrome, Firefox, and Edge.
<b>Test Number(s)</b>	2

Table 10

<b>Number</b>	3
<b>Priority</b>	5
<b>Description</b>	All questions entered by the teacher or administrator shall appear on the output survey.
<b>Test Number(s)</b>	3

Table 11

<b>Number</b>	4
<b>Priority</b>	5
<b>Description</b>	All data stored in the program's database shall be valid.
<b>Test Number(s)</b>	4

Table 12

<b>Number</b>	5
<b>Priority</b>	5
<b>Description</b>	All collected survey data shall not be alterable.
<b>Test Number(s)</b>	5

Table 13

<b>Number</b>	6
<b>Priority</b>	4
<b>Description</b>	Teachers shall not be able to access data of courses other than their own.
<b>Test Number(s)</b>	6

Table 14

<b>Number</b>	7
<b>Priority</b>	3
<b>Description</b>	The mean time between failures should be at least 60 minutes.
<b>Test Number(s)</b>	7

Table 15

<b>Number</b>	8
<b>Priority</b>	5
<b>Description</b>	Students shall have no access to any data stored by the program.
<b>Test Number(s)</b>	8

Table 16

<b>Number</b>	9
<b>Priority</b>	5
<b>Description</b>	All survey responses (except signed comments) shall be anonymous.
<b>Test Number(s)</b>	9

Table 17

<b>Number</b>	10
<b>Priority</b>	2
<b>Description</b>	The software should scale to at least three universities, 1000 courses per semester, 1000 teachers per university, and 500 students per course.
<b>Test Number(s)</b>	10

Table 18

<b>Number</b>	11
<b>Priority</b>	1
<b>Description</b>	The software should not exceed 500 MB in size.
<b>Test Number(s)</b>	11

Table 19

<b>Number</b>	12
<b>Priority</b>	4
<b>Description</b>	The software's source code shall be open-source and shall use a GPLv2 license.
<b>Test Number(s)</b>	12

Table 20

<b>Number</b>	13
<b>Priority</b>	4
<b>Description</b>	The licensing requirements of any non-original code shall be met.
<b>Test Number(s)</b>	13



Table 21

<b>Number</b>	14
<b>Priority</b>	4
<b>Description</b>	The software shall meet UMaine AFUM requirements.
<b>Test Number(s)</b>	13

### 3.1 Tests

These are the tests that will verify the non-functional requirements:

1. The tester attempts to run the program on the last versions of Windows, Mac, Linux, iOS, and Android. The test is successful only if all five use cases execute properly on each operating system.
2. The tester attempts to run the program on the last versions of Safari, Chrome, Firefox, and Edge. The test is successful only if all five use cases execute properly on each browser.
3. The tester inputs five diverse questions into a mock survey and sends the survey to a mock student. The test is successful only if the mock student can follow the link to a survey showing those five questions.
4. The tester inputs a course, questions preset, and survey. The test is successful only if all data for the course, preset, and survey are present in the database, and if that data correctly follows the database schema.
5. The tester inputs the same course, preset, and survey as in test 4. The test is successful only if any function in the software does not change the survey responses stored in the database.
6. The tester creates three users for the evaluation program. Each user inputs a different course, questions preset, and survey. The test is successful if the users cannot view nor modify each other's data.
7. The tester runs an automatic script that simulates several hours of using the software. The test is successful only if the average time between two errors in the software is at least 60 simulated minutes.
8. The tester inputs a course, present, and survey as a teacher, then poses as someone unauthorized to use the software. In a successful test, the tester cannot view nor modify any data present in the database.
9. Five testers complete a survey sent by the software, and a sixth tester uses the software afterwards. The test is successful only if the sixth tester cannot find the identities of anyone who gave a response meant to be anonymous.
10. The tester runs an automatic script that sets up the software with three colleges, 1000 courses per college, 1000 teachers per university, and 500 students per course. The test is successful if all five use cases execute properly for all courses.
11. The tester sees the info for the folder containing the code. The folder size is at most 500 megabytes if the test passes.
12. Ten computer users attempt to access the source code on GitHub. The test is successful if all users can do so.
13. The tester reads the GNU General Public License and University of Maine AFUM requirements. The test is successful if the software meets all terms in both the license and AFUM requirements.

## 4 User Interface

See “User Interface Design Document for the College Course Evaluation System.”

## 5 Deliverables

The following lists the estimated date and format that each submission will be delivered:

Table 22

Submission	Date of Delivery	Format
System Requirements Specification	10/29/2018	Hard Copy
System Design Document	11/16/2018	Hard Copy
User Interface Design Document	11/30/2018	Hard Copy
Administrator Manual	April 2019	Hard Copy
User Manual	May 2019	Hard Copy
System Requirements Specification	May 2019	Electronic
System Design Document	May 2019	Electronic
User Interface Design Document	May 2019	Electronic
Administrator Manual	May 2019	Electronic
User Manual	May 2019	Electronic
Source Code	May 2019	Electronic
Web Link to Program	May 2019	Electronic

## 6 Open Issues

The most significant issue our team has relates to the non-anonymous student responses. In a course evaluation survey, a student has the option to add a signed comment to be stored in an instructor’s file. However, the LimeSurvey software does not store the identities of survey respondents. We need to find a way to collect the signed comment along with the survey, while ensuring that the signature is authentic.

There are several more minor issues that need to be resolved. We have not decided what information about each course should be stored in the database. We do not know what format to use for storing the survey data. We should also know which question presets and which surveys’ results are visible to each user. Another issue is how often to notify students about their survey. Finally, we need to know how to authenticate a student who wants to submit a non-anonymous response.

The team projects that all of the above issues will be resolved by the end of November.

## A Agreement Between Customer and Contractor

This page shows that all members of Team EVAL and the client, Harlan Onsrud, have agreed on all the information in the system requirements specification. By signing this document, Team EVAL and Dr. Onsrud agree on the goals and scope of the project, each use case and requirement, the tests for each requirement, and how the deliverables will be sent.

The team will follow a process in the case that the requirements specification is changed after we sign it. First, the team writes a rough draft of the changes to be made to the document. Second, all team members and Harlan Onsrud will sign the document agreeing to the changes. Finally, the changes are made to the final copy of the specification.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
Jovon Craig	_____	_____
Sam Elliott	_____	_____
Robert Judkins	_____	_____
Stanley Small	_____	_____
Harlan Onsrud	_____	_____
Customer Comments:	_____	
_____		

# B Team Review Sign-off

This page shows that all members of Team EVAL have reviewed the system requirements specification and agreed on its content. By signing this document, the team members agree on the goals and scope of the project, each use case and requirement, the tests for each requirement, and how the deliverables will be sent. There is nothing in the document that is a source of contention.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
<b>Jovon Craig</b>	_____	_____
Comments:	_____	
_____		
<b>Sam Elliott</b>	_____	_____
Comments:	_____	
_____		
<b>Robert Judkins</b>	_____	_____
Comments:	_____	
_____		
<b>Stanley Small</b>	_____	_____
Comments:	_____	
_____		

## C Document Contributions

Stanley Small created the document, added the system requirements specification template to the document, and formatted the text to be compatible with LaTeX. He added the log-in use case to the requirements and the team logo to the table of contents. He was also an active participant in discussing and writing down requirements during meetings with the client. Stan contributed approximately 20 percent of the document.

Jovon Craig wrote five of the use cases in the functional requirements, all non-functional requirements, and a draft of the requirement tests. He also wrote the purpose of the document, the purpose of the product, the deliverables section, and the three appendices. Jovon contributed about 55 percent of the document.

Sam Elliott worded the non-functional requirements, wrote the section explaining the product scope, and created the use case diagram that describes the system. He also added the “publish survey” use case and two tests for the functional requirements. Sam contributed about 20 percent of the document.

Robert Judkins made revisions to the tests for the functional requirements. He contributed about 5 percent of the document.

## C System Design Document

# System Design Document

Course Evaluation System

Team EVAL

Jovon Craig, Sam Elliott, Robert Judkins, and Stanley Small

Client: Dr. Harlan Onsrud

November 16, 2018

University of Maine - Fall of 2018 - COS 397  
Instructor: Professor Terry Yoo



# Course Evaluation System

## System Design Document

### Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose of This Document . . . . .	2
1.2	References . . . . .	2
<b>2</b>	<b>System Architecture</b>	<b>2</b>
2.1	Architectural Design . . . . .	2
2.2	Decomposition Description . . . . .	3
2.2.1	Endpoint Descriptions . . . . .	5
<b>3</b>	<b>Persistent Data Design</b>	<b>5</b>
3.1	Database Descriptions . . . . .	5
3.2	File Descriptions . . . . .	6
<b>4</b>	<b>Requirements Matrix</b>	<b>7</b>
<b>A</b>	<b>Agreement Between Customer and Contractor</b>	<b>8</b>
<b>B</b>	<b>Team Review Sign-off</b>	<b>9</b>
<b>C</b>	<b>Document Contributions</b>	<b>10</b>
<b>D</b>	<b>Example Question Selection Form</b>	<b>11</b>

# 1 Introduction

Team EVAL is creating a system to more efficiently create and distribute post-semester teaching evaluations. The system will be built to interface with LimeSurvey, a popular open-source survey application.

## 1.1 Purpose of This Document

This system design document gives an overview of the structure and planned implementation of our course evaluation system. The first section describes the system architecture, how the components function, and how the components relate to each other. The second section covers the design of the data, detailing the database schema and the properties of the files that are used by the system. The document's third section has a table that lists the components that fulfill each functional requirement.

This document is intended for the development team, the product client, Dr. Harlan Onsrud, and potential users of the system. Team EVAL needs this document to ensure that the product works as intended. Dr. Onsrud needs it to know that the program that he desires will be fully realized. The document also helps the software's users in that they can learn more about the functions and architecture of the evaluation system.

## 1.2 References

Craig, J., Elliott, S., Judkins, R., & Small, S. 29 October 2018. *System Requirements Specification*.

Craig, J., Elliott, S., Judkins, R., & Small, S. 30 November 2018. *User Interface Design Document*.

Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Boston: Addison-Wesley.

Using OAuth 2.0 to Access Google APIs — Google Identity Platform — Google Developers. Google, Google, 12 Nov. 2018, [developers.google.com/identity/protocols/OAuth2](https://developers.google.com/identity/protocols/OAuth2).

Onsrud, H. "Example Question Selection Form." See Appendix D.

# 2 System Architecture

## 2.1 Architectural Design

Figure 1 is a high-level abstraction of the proposed system architecture. With this figure, we aim to communicate what components are in the system and the API allowing the components to communicate.

Figure 1: Component diagram of the system (left); sequence diagram of Google OAuth 2.0 (right)

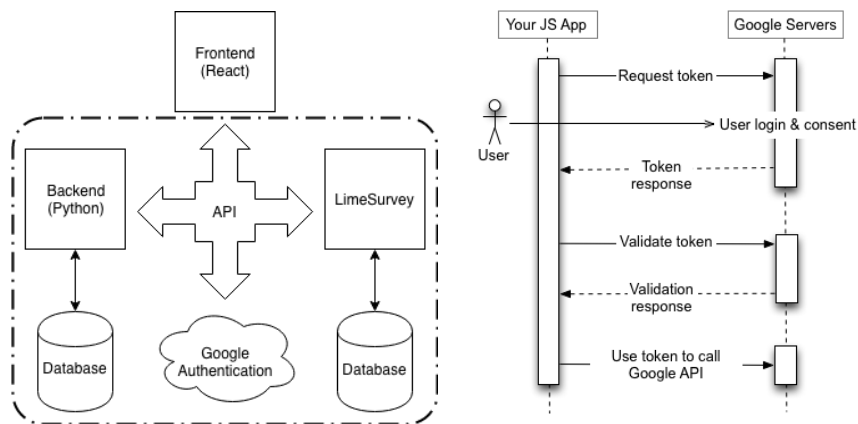
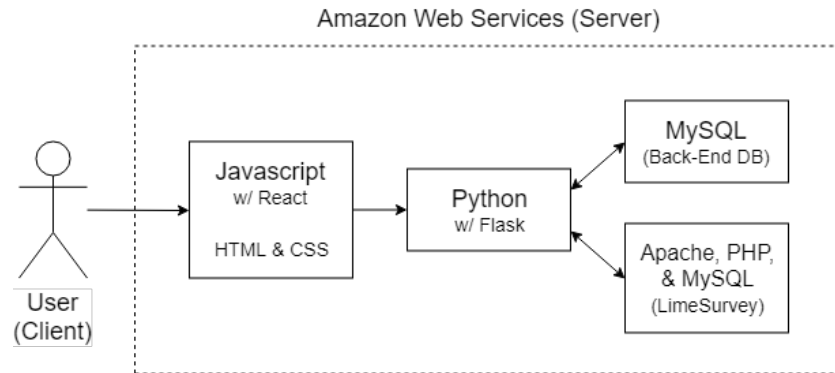




Figure 2 is a technology architecture diagram that shows the languages and frameworks that the team will use to make the system.

Figure 2: Technology diagram of the system



LimeSurvey (version 3.15.3+181108) is running on an Apache web server that uses PHP version 5.6. This survey software maintains a database running on MySQL 5.5.61. The team will use Amazon Web Services to host the evaluation system on the Web at [teachingevaluations.org](http://teachingevaluations.org), providing a scalable solution to fluctuating demand and peak use after the semester.

The system consists of three major components, which communicate via an API. The front end, which handles the user interface, and displays the visuals and data sent from the API. It will be written in JavaScript using the React framework, along with HTML and CSS. The back end communicates with the databases, which are hosted by the same database management system. It will be written in Python, and its endpoints will use the Flask library. The database stores survey templates and responses using MySQL.

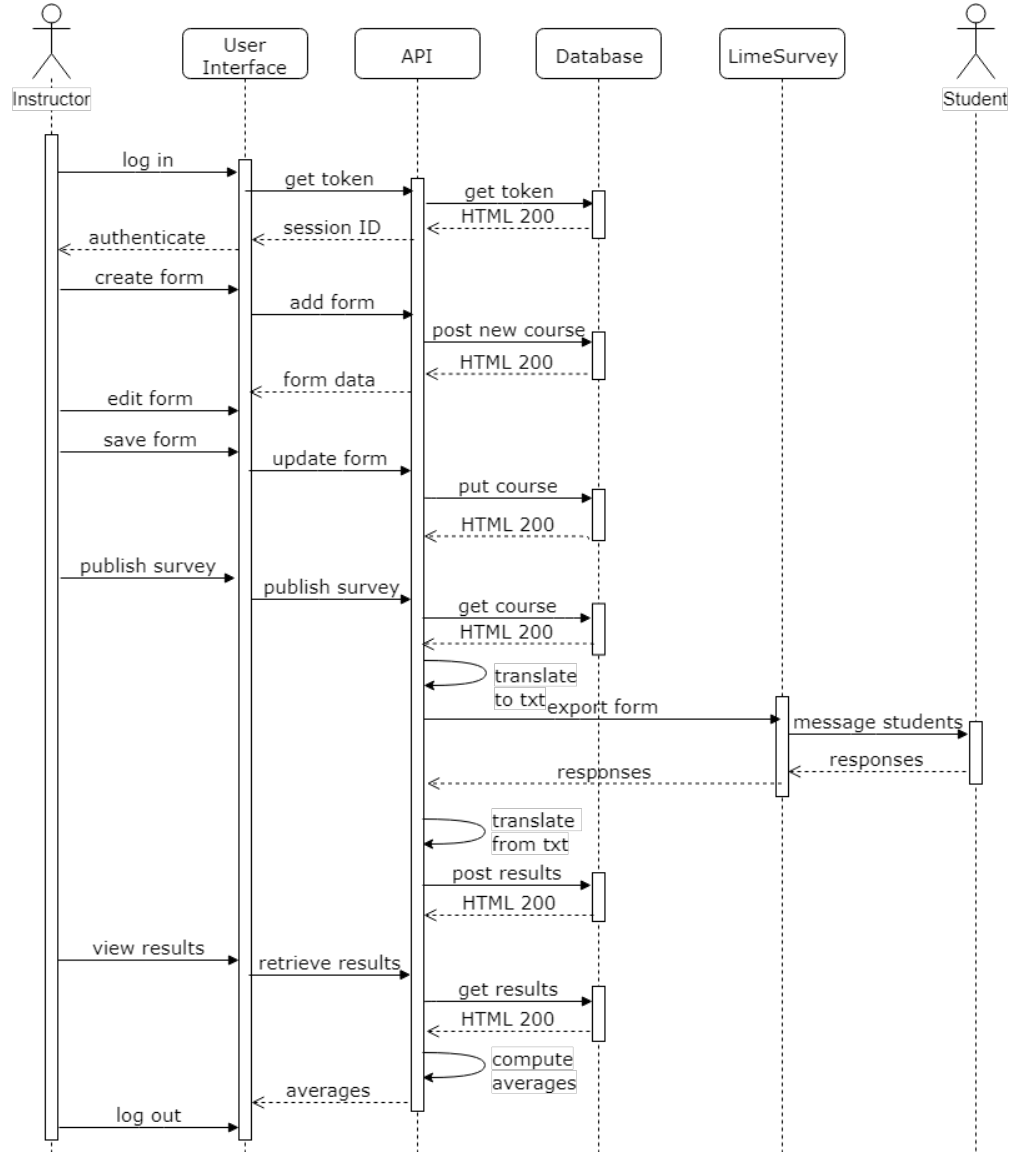
A single API will connect the front end to the back end and the back end to LimeSurvey and our database. A REST interface connects the front end to the back end using endpoints. Separate functions will utilize the LimeSurvey API so that the back end can access the LimeSurvey database. LimeSurvey is outside the scope of the project and our system will not alter LimeSurvey in any way. A user will have no contact with LimeSurvey and will only communicate with the front end.

Google OAuth 2.0 will be used to authenticate users with their Google accounts. Each student at the University of Maine is issued a Google Apps account e-mail which can be used to log in to the teaching evaluation system. As shown in the right side of Figure 1, the application will request a token from Google. The application will then store this token in a cookie. When a page is refreshed, the token is validated on Google's servers. Google OAuth 2.0 can also provide information such as the user's name and e-mail.

## 2.2 Decomposition Description

Figure 3 abstracts the major functions that are expected to be in the system. This sequence diagram, shown on the next page, illustrates a typical session and is meant to communicate a more detailed view of the components and their relationships. The "Database" refers to the database that we will be implementing to store survey templates and responses.

Figure 3: Sequence diagram of an example session



To begin using the course evaluation system, the instructor must first log in with a username and password (“log in”). The API retrieves the user’s token from the database to confirm that the password is correct (“get token”, “authenticate”). Next, the instructor creates an evaluation form to send to students (“create form”). The API adds an empty course to the database (“add form”, “post new course”). The instructor then edits the survey with the appropriate info and saves the form (“edit form”, “save form”). Upon saving, the API updates the database with the entered information (“update form”, “put course”).

With an evaluation form completed, the instructor can then publish it online (“publish survey”). The API retrieves the course information from the database to translate the survey form into a tab-separated .txt file (“get course”, “translate to txt”). It then sends the .txt file to LimeSurvey and tells the survey software to message the students through e-mail (“export form”, “message students”). The API retrieves the students’ responses, translates them from another tab-separated .txt file, and stores them into the database (“translate from txt”, “post results”). Finally, the instructor can view the survey results, with the help of the API getting the results from the database and finding their averages (“retrieve results”, “get results”, “compute averages”).

### 2.2.1 Endpoint Descriptions

The following table briefly describes the API’s endpoints, which query data from the back-end database:

Table 1: API endpoint descriptions

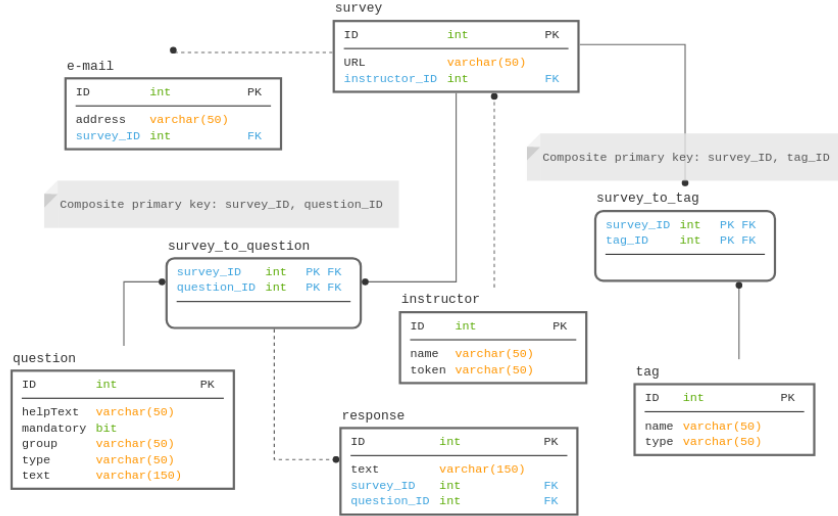
Type	Name	Description
GET	/login	retrieves a token for a certain authentication key
POST	/create_user	adds a user to the database
GET	/courses	retrieves a list of all courses
GET	/course	retrieves the info for a given course
PUT	/course	updates the info for a given course
DELETE	/course	removes a given course
POST	/new_course	creates a new course
GET	/results	retrieves a list of survey results for a given course
POST	/results	updates a list of survey results for a given course

## 3 Persistent Data Design

### 3.1 Database Descriptions

Our database will include tables that store information about courses, survey questions, and survey responses. A diagram of the database schema is shown in figure 4 on the next page:

Figure 4: Schema diagram of the back-end database

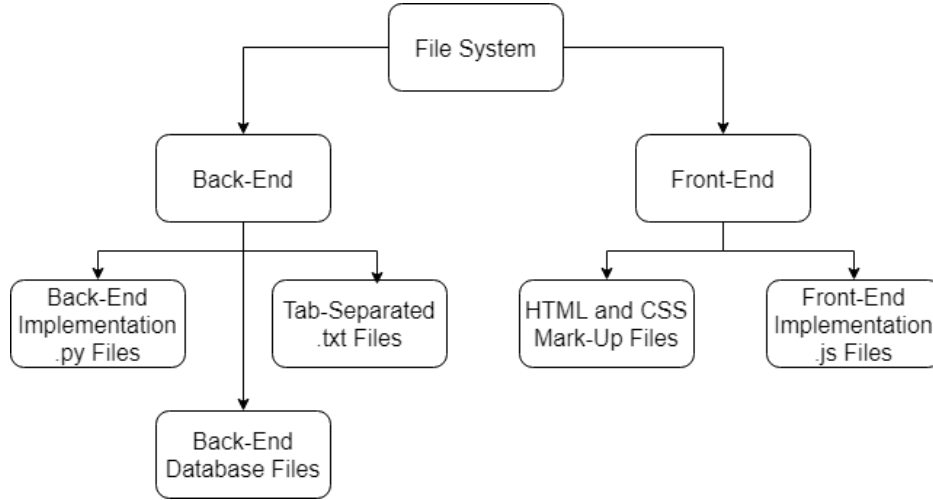


The database contains six object tables and two relationship tables. The **Survey** table contains survey IDs, the surveys' URLs (generated by LimeSurvey) and associated instructor IDs. The e-mail addresses to send the surveys to are stored in the **E-mail** table. The **Instructor** table contains the names of the instructors along with authentication tokens generated by Google OAuth. The **Tag** table contains additional information (e.g. course names) about the surveys. These tags are used to categorize and search surveys. The **Question** table contains all questions to be entered into the surveys, along with help text and the type of question. The **Response** table includes students' responses to certain questions. The relationship tables, **Survey-to-Question** and **Survey-to-Tag**, link the surveys to their appropriate questions and tags.

### 3.2 File Descriptions

The system requires several pieces of data, including database files, front-end markup, and LimeSurvey files, to run as intended. A file structure diagram is shown in figure 5 on the next page:

Figure 5: Diagram of the files used by the system



The back-end database has a file to keep track of all course information, survey forms, and survey responses that users have input. The schema in the previous section shows the fields and types in that file. To reduce the risk of a breach, any survey results in the database get removed by the system 60 days after viewed by the instructor.

The front end will be implemented in JavaScript, so our file structure will include .js files that operate the front end. There are also mark-up files, which specify the look and feel of the front end. The back-end will be implemented in Python and our file structure will include .py files that operate the back end. The JavaScript and Python files are static and permanent, and thus do not require maintenance.

Lastly, the system briefly stores LimeSurvey files in the form of tab-separated text documents. They either contain survey form data or student responses. The form data files are deleted immediately after they are sent to LimeSurvey, and the response data files are deleted immediately after they are stored in the database.

## 4 Requirements Matrix

The following table lists the functions in our system, as shown in the sequence diagram, that meet each functional requirement given in the system requirements specification:

Table 2: Requirements matrix

Use Case Number	Use Case Name	System Component(s)
1	Log on	log in, get token
2	Create survey	create form, add form, post new course
3	Edit survey	edit form, save form, update form, put course
4	Publish survey	publish survey, get course, translate to txt, export form, message students
5	View survey results	view results, retrieve results, get results, compute averages

## A Agreement Between Customer and Contractor

This page shows that all members of Team EVAL and the client, Harlan Onsrud, have agreed on all the information in the system design document. By signing this document, Team EVAL and Dr. Onsrud agree on the system's architecture, components, relations between the components, database schema, required files, and file descriptions.

The team will follow a process in the case that the design document is changed after we sign it. First, the team writes a rough draft of the changes to be made to the document. Second, all team members and Harlan Onsrud will sign the document agreeing to the changes. Finally, the changes are made to the final copy of the document.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
Jovon Craig	_____	_____
Sam Elliott	_____	_____
Robert Judkins	_____	_____
Stanley Small	_____	_____
Harlan Onsrud	_____	_____
Customer Comments:	_____	
_____		

# B Team Review Sign-off

This page shows that all members of Team EVAL have reviewed the system design document and agreed on its content. By signing this document, the team members agree on that all information about the system’s architecture and design are accurate. There is nothing in the document that is a source of contention.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
<b>Jovon Craig</b>	_____	_____
Comments:	_____	
_____		
<b>Sam Elliott</b>	_____	_____
Comments:	_____	
_____		
<b>Robert Judkins</b>	_____	_____
Comments:	_____	
_____		
<b>Stanley Small</b>	_____	_____
Comments:	_____	
_____		

## C Document Contributions

Stanley Small included a template of the appendices and he wrote a draft of the title page, architectural design section, and database description. He also added the component diagram along with the Google OAuth sequence diagram, and helped with the database schema. Stan contributed approximately 30 percent of the document.

Jovon Craig wrote the purpose of the document, references, decomposition description, file descriptions, and requirements matrix. He made revisions to the title page, architectural design section, and database description. He also added the sequence diagram and file diagram, revised the technology diagram, and made a draft of the database schema. Jovon contributed about 40 percent of the document.

Sam Elliott created the technology diagram rough draft, and made revisions to the sequence diagram and the file diagram and added many of them to the document. He revised the architecture design section and the file descriptions section, as well as general changes based on feedback from the SRS and Dr. Onsrud. He was the main point of contact with Dr. Onsrud for reviewing the document and organizing meetings. Sam contributed about 20 percent of the document.

Robert Judkins converted the document to LaTeX and helped with front-end design. Robert contributed about 10 percent of the document.



## D Example Question Selection Form

### APPENDIX A SELECTION OF QUESTIONS AND CONTENT FOR INCLUSION ON A WEB-BASED STUDENT EVALUATION OF TEACHING FORM

#### INSTRUCTOR OR ACADEMIC UNIT SELECTION OF QUESTIONS FOR USE IN A WEB-BASED FORM:

If you want information on this form auto-filled from a previous submission that you may then edit, please **SELECT** (Note to Team: **SELECT** leads to a pulldown menu that includes all previous submission forms. When selected, all information would be copied from a previous form to the current form below except as noted.)

Course Designator (e.g. MUS)	_____
Course Number (e.g. 200)	_____
Course Section (e.g. 001)	_____
Course Title (e.g. Ballroom Dance)	_____
Is this a graduate course?	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> (bull's eye choices)
Semester and Calendar Year (e.g. Fall 2019)	_____ (don't auto populate)_
Name of Faculty Unit (e.g. School of Music)	_____
Name of College (e.g. Liberal Arts)	_____
Name of University (e.g. Univ of Maine)	_____
Last Name of Instructor (e.g. Smith)	_____ (need later in reporting to alphabetize by last name)
First Name of Instructor (e.g. Elizabeth)	_____
Instructor Email	_____
Instructor Phone	_____
Full Name of Course Evaluation Administrator	_____
Email of Course Evaluation Administrator	_____
Beginning Date of Assessments	_____ (use calendar pulldown) (don't auto populate)_
Time of Day for Any and All Mailings	_____ (use time select)_____ (don't auto populate)_
Closing Date	_____ (use calendar pulldown) (don't auto populate)_

#### NOTE TO CAPSTONE TEAM:

Variable names from some of the entries above include: {COURSEDESIGNATOR}, {COURSENUMBER}, {COURSETITLE}, {ADMINNAME}, {ADMINEMAIL}, {BEGINNINGDATE}, {CLOSINGDATE}, {TIME}. Some other variable names likely to be used coming from the student roll upload include: {FIRSTNAME}, {LASTNAME}, {EMAIL}.

# UNIVERSITY OF MAINE STUDENT EVALUATION OF TEACHING

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>THE INSTRUCTOR</b>			
How prepared was the instructor for class?	often unprepared; well prepared		
How clearly were the objectives of the course presented?	unclear; very clear		
How enthusiastic was the instructor about the subject?	very little; very much		
How clearly did the instructor present concepts, principles and theories?	unclear; very clear		
How much were you encouraged to think for yourselves?	very little; very much		
How concerned was the instructor for the quality of student learning?	unconcerned; very concerned		
Did the instructor show respect for the questions and opinions of the students?	rarely; always		
Did the instructor ensure an environment of respect for all groups of people in the classroom?	rarely; always		
Did the instructor inspire confidence in his/her knowledge?	very little; very much		
<b>Overall, how would you rate the instructor?</b>	poor; excellent		
Please list further questions to be added regarding the instructor, if any, indicating a 1-5 rating scale (i.e. 5 is best)			
(a)			
(b) Online creator should be able to add up to 15 questions			
<b>THE COURSE</b>			
Were class meetings profitable and worth attending?	rarely; always		
What is your overall rating of the primary readings?	poor; excellent		
How much did this course challenge you intellectually?	very little; very much		
How much did you learn from this course?	very little; very much		
<b>What is your overall rating of this course?</b>	poor; excellent		
Please list further questions to be added regarding the course, if any, indicating a 1-5 rating scale (i.e. 5 is best)			
(a)			
(b) Online creator should be able to add up to 15 questions			
<b>STUDENT WORK ASSESSMENT</b>			
Did the instructor let you know what was expected on the assessments (exams, assignments, projects, papers, etc.)?	unclear; very clear		
Did the assessments reflect the important aspects of the course?	rarely; always		
How fair were the grading procedures?	unfair; completely		
Overall, how would you rate the assessment process (exams, assignments, projects, papers, etc.)?	poor; excellent		
Please list further questions to be added regarding the assessment, if any, indicating a 1-5 rating scale (i.e. 5 is best)			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **laboratory questions** were copied from a previous University of Maine approved longer bubble form.

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>THE LABORATORY EXPERIENCE</b>			
Did this course have one or more regularly scheduled laboratory sessions? (If the student answer is yes, the following questions appear)			
How much did the laboratory experience contribute to your learning in this course?	very little, very much		
<b>Overall, how would you rate the laboratory experience?</b>	poor; excellent		
What was done particularly well in the laboratory experience?	----		
How could the laboratory experience be improved?	----		
Please list further lab questions to be added, if any, and a rating scale			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **teaching assistant questions** were copied from a previously used UMaine longer bubble form.

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>THE TEACHING ASSISTANT</b>			
Was there a teaching assistant supporting this course? (If answer is yes, the following questions appear)			
If there was more than one teaching assistant for the course, please name the TA you are evaluating. (open ended response)			
How much did the teaching assistant contribute to your learning in this course?	very little, very much		
How concerned was the teaching assistant for the quality of student learning?	unconcerned; very concerned		
Did the teaching assistant show respect for the questions and opinions of students?	rarely; always		
Did the teaching assistant inspire confidence in his/her knowledge?	very little; very much		
Would you want to have this teaching assistant in the future in another course?	definitely not, definitely yes		
Would you recommend this teaching assistant to assist in this course in the future?	definitely not, definitely yes		
<b>Overall, how would you rate the teaching assistant?</b>	poor; excellent		
Name something the teaching assistant did particularly well.	----		
Name something the teaching assistant could do better in the future.	----		
Please list further TA questions to be added, if any			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **online component questions** were copied from the current UMaine Augusta distance education form

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>ONLINE COMPONENT ASSESSMENT</b>			
Did you take this course as a distance learning student? (If answer is yes, the following questions appear)			
Please indicate the primary online modality used with the course. [only one selection allowed] <radio> Sessions were primarily <u>synchronous</u> in that I was typically required or highly encouraged to interact live online with the instructor and other students in most or all sessions. <radio> Sessions were primarily <u>asynchronous</u> in that I primarily viewed recorded video lectures or class sessions on my own schedule and had only one or fewer opportunities per week to interact live online with the instructor or other students. <radio> Sessions were approximately an equal mix of <u>synchronous</u> and <u>asynchronous</u> interactions. <radio> Other. Describe: <short text box>	----		
The online modality used with the course was well suited to my needs.	strongly disagree, strongly agree		
There was adequate opportunity for me to interact with the instructor.	strongly disagree, strongly agree		
There was adequate opportunity for me to interact with other students.	strongly disagree, strongly agree		
The online technologies used in this course worked the way they were supposed to.	strongly disagree, strongly agree		
The communication tools were easy to use (email, assignment delivery, exam delivery or proctoring, chat, blog, teleconferencing, online courseware, web, etc.).	strongly disagree, strongly agree		
Technology support was there if I needed it.	strongly disagree, strongly agree		
The online experience was well-suited to the way I like to learn.	strongly disagree, strongly agree		
Which statement best characterizes your belief after having taken this course? [only one selection allowed] <radio> I learned MUCH LESS in this online class than I probably would have in the traditional format. <radio> I learned A LITTLE LESS in this online class than I probably would have in the traditional format. <radio> I learned ABOUT THE SAME in this online class than I probably would have in the traditional format. <radio> I learned A LITTLE MORE in this online class than I probably would have in the traditional format. <radio> I learned MUCH MORE in this online class than I probably would have in the traditional format.	----		

Which statement best characterizes your belief about the grade you expect to receive in this course? [only one selection allowed]	----		
<radio> I worked MUCH LESS for my grade in this online class than I probably would have had to in the traditional format.			
<radio> I worked A LITTLE LESS for my grade in this online class than I probably would have had to in the traditional format.			
<radio> I worked ABOUT THE SAME for my grade in this online class than I probably would have had to in the traditional format.			
<radio> I worked A LITTLE HARDER for my grade in this online class than I probably would have had to in the traditional format.			
<radio> I worked MUCH HARDER for my grade in this online class than I probably would have had to in the traditional format.			
Please identify an e-learning aspect of the course that you found particularly valuable or beneficial	----		
Please identify an e-learning aspect of the course that could be improved.	----		
Please list further questions to be added, if any			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **open-ended questions** were copied from the current University of Maine bubble form.

Questions	1-5 Rating Scale	Include question? (If yes, indicate with ✓)	Make response mandatory? (If yes, indicate with ✓)
<b>OPEN ENDED QUESTIONS</b>			
Please identify the aspects of this course that were of most value to you.	----		
Please mention at least one additional topic or component that you would like to see included in this course.	----		
Please make any additional comments that you desire to make about the course instructor, materials or pedagogy.	----		
Please list further open-ended questions to be added, if any			
(a)			
(b) Online creator should be able to add up to 15 questions			

#### Additional Option for Mandatory Questions

For questions in the tables above that you have designated as mandatory, should the last response option for possible selection on all of these questions be “decline to respond? (If “yes”, an option 6 will be added) Yes \_\_\_ No \_\_\_

#### CLASS ROLL

**Please insert in the field below the first name, last name and email of each student in the class.** The information for each student should appear in a separate row and be separated by commas. (e.g. Mary, Smith, marysmith@gmail.com) You may cut and paste into the window at your option.

<window for entry – do not autopopulate in future subsequent uses of the form>

Alternatively, upload a cvs file with the content in the first row of the file being firstname, lastname, email. **UPLOAD**

#### EMAILS TO BE SENT TO STUDENTS

**Initial Email invitation to Participate**

Although we recommend that you do not change the following email text, you may edit the *Invitation to Participate* as appropriate for your purposes if needed. Do NOT change any item listed as a {VARIABLE} because this will cause an error in your submission that you will be forced to correct prior to successful submission. This email will be sent to each student on the begin date at the time you specified above.

<window for text – Team, Please insert from below the suggested text for DEFAULT INITIAL INVITATION TO STUDENTS>

**Do you want one or more reminder emails sent to students who have yet to respond after a few days?** Yes \_\_\_ No \_\_\_  
(If the answer is yes, the following statements and questions appear)

#### **Reminder Emails**

You may send up to three reminder emails to those students who have yet to complete the teaching evaluation. The text for all reminder emails will be identical.

Although we recommend that you do not change the following email text, you may edit the *Reminder Emails* to students as appropriate for your purposes if needed. Do NOT change any item listed as a {VARIABLE} because this will cause an error in your submission that you will be forced to correct prior to successful submission.

<window for text – Team, Please insert from below the suggested text for DEFAULT REMINDER TO STUDENTS>

Reminder emails are sent spaced three days apart until the closing date is reached unless you specify a longer interval.

Please send reminder emails at intervals of <insert number of 3 or above> days.

Only a maximum of three reminders will be sent regardless of the spacing. Reminder mailings are sent at the same time of day as the initial invitation mailing.

#### **Confirmation Message**

Although we recommend that you do not change the following email text, you may edit the *Confirmation Message* reminder to students as appropriate for your purposes if needed. Do NOT change any item listed as a {VARIABLE} because this will cause an error in your submission that you will be forced to correct prior to successful submission.

<window for text – Team, Please insert from below the suggested text for DEFAULT CONFIRMATION>

**Please double check all information provided above. Once you click SUBMIT you may not make further changes unless you receive an error notification. You will receive a summary of the student responses by email at the time and date you designated for the teaching evaluation process to end.**

IMPORTANT: You have NOT successfully completed your assessment submission until you press the SUBMIT button below AND you are taken to a web page that states your submission has been successful. If you are returned to an earlier portion of this form, complete all information items now highlighted in red and then click SUBMIT again.

#### **SUBMIT**

+++++

**DEFAULT INITIAL INVITATION TO STUDENTS FOR EDITING BY THE  
INSTRUCTOR/ADMINISTRATOR USER**

+++++

Subject: Invitation to Complete Evaluation for {COURSEDESIGNATOR} {COURSENUMBER} {COURSETITLE}

Dear {FIRSTNAME},

Please complete the teaching and course evaluation for {COURSEDESIGNATOR} {COURSENUMBER}  
{COURSETITLE}.

This student evaluation of teaching is completely anonymous unless you purposefully identify yourself in response to one of the questions. The software system will send you automatic reminders every few days until you complete the evaluation.

To respond, simply click the link at the end of this message.

Sincerely,  
{ADMINNAME}

-----  
Click here to complete the teaching and course evaluation:  
{SURVEYURL}

{ADMINNAME} ({ADMINEMAIL})

++++++  
**DEFAULT REMINDER TO STUDENTS FOR EDITING BY THE INSTRUCTOR/ADMINISTRATOR USER**  
– TO BE SENT AT REGULAR INTERVAL UNTIL STUDENT RESPONDS, MAXIMUM OF THREE IS REACHED,  
OR CLOSING DATE IS REACHED  
++++++

Subject: Reminder to Complete Evaluation for {COURSEDESIGNATOR} {COURSENUMBER} {COURSETITLE}

Dear {FIRSTNAME},

Recently we invited you to complete a teaching evaluation for “{COURSEDESIGNATOR} {COURSENUMBER} {COURSETITLE}”. We note that you have not yet completed the evaluation, and wish to remind you that it is still available should you wish to take part.

To participate, please click on the link below.

The deadline for completing the evaluation is {CLOSINGDATE} at {TIME}.

Your participation is extremely important to the improvement of teaching and courses at the {UNIVERSITYNAME}

Sincerely,  
{ADMINNAME}

-----  
Click here to complete the teaching and course evaluation:  
{SURVEYURL}

**NOTE TO TEAM: SHOULD THE FOLLOWING GREEN OPTION BE INCLUDED AND IMPLEMENTED?**  
If you do not want to participate in the student evaluation process and don't want to receive any more email reminders, please click the following link:  
{OPTOUTURL}

Note: Such an option is required in many instances by “Do Not Spam” federal legislation but would not be required typically in a university mailing to your own students. Implement as good practice regardless?

{ADMINNAME} ({ADMINEMAIL})

++++++  
**DEFAULT CONFIRMATION**  
++++++

Subject: Completion of Course Evaluation for {COURSENUMBER} {COURSETITLE}

Dear {FIRSTNAME},

This email is to confirm that you have completed a student evaluation of {COURSEDESIGNATOR}  
{COURSENUMBER} {COURSETITLE} at the {UNIVERSITYNAME}. Your response has been saved. Thank you for  
participating.

If you have any further questions about this email, please contact {ADMINNAME} at {ADMINEMAIL}.

Sincerely,

{ADMINNAME}



## D User Interface Design Document

# User Interface Design Document

## Course Evaluation System

Team EVAL

Jovon Craig, Sam Elliott, Robert Judkins, and Stanley Small

Client: Dr. Harlan Onsrud

November 30, 2018

University of Maine - Fall of 2018 - COS 397  
Instructor: Professor Terry Yoo



# Course Evaluation System

## User Interface Design Document

### Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose of This Document . . . . .	2
1.2	References . . . . .	2
<b>2</b>	<b>User Interface Standards</b>	<b>2</b>
<b>3</b>	<b>User Interface Walkthrough</b>	<b>4</b>
<b>4</b>	<b>Data Validation</b>	<b>15</b>
<b>A</b>	<b>Agreement Between Customer and Contractor</b>	<b>16</b>
<b>B</b>	<b>Team Review Sign-off</b>	<b>17</b>
<b>C</b>	<b>Document Contributions</b>	<b>18</b>
<b>D</b>	<b>Example Question Selection Form</b>	<b>19</b>
<b>E</b>	<b>Example Results Display</b>	<b>27</b>

# 1 Introduction

## 1.1 Purpose of This Document

This user interface design document is an overview of the graphics and layouts shown to the users of our course evaluation system. The first section, the user interface standards, describes the general features of the graphics, such as layouts and components, that are common to all screens in the interface. The second section, the user interface walkthrough, includes a “navigation diagram” of the order in which screens appear, as well as complete wireframes of each screen. The document’s third section gives the data items typically entered in the user interface and how they are formatted.

This document is intended for the development team, the product client, Dr. Harlan Onsrud, and potential users of the system. Team EVAL needs this document to properly implement the user interface in code. Dr. Onsrud also needs it to verify that the program’s appearance looks appropriate for universities. Lastly, the document helps the software’s users in that it serves as a guide for how to use the software.

## 1.2 References

Craig, J., Elliott, S., Judkins, R., & Small, S. 29 October 2018. *System Requirements Specification*.

Craig, J., Elliott, S., Judkins, R., & Small, S. 16 November 2018. *System Design Document*.

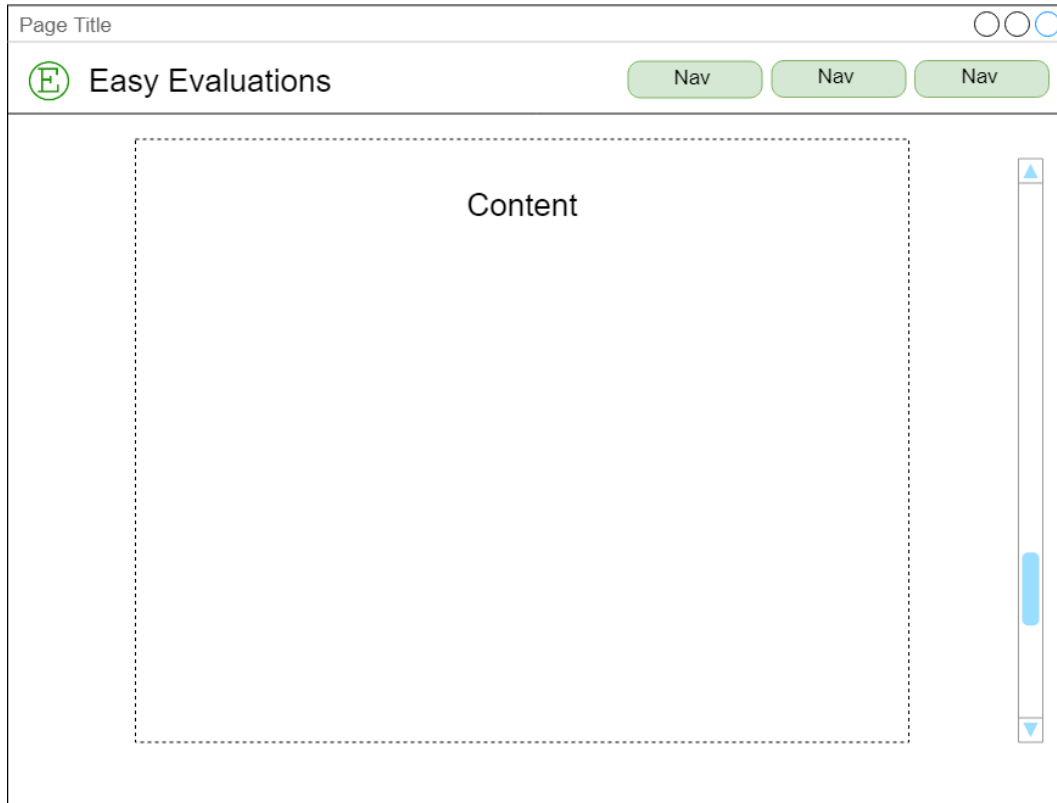
Onsrud, H. “Example Question Selection Form.” See Appendix D.

Onsrud, H. ”Report for Professor: Roy Turner” See Appendix E.

# 2 User Interface Standards

The interface of the course evaluation system is standardized; several components are present in multiple screens. Figure 1, shown on the next page, is the overall screen layout of our course evaluation system. It shows the general areas and components of the screens in the user interface. Not all screens follow the exact overall layout.

Figure 1: Overall layout of a screen in the UI

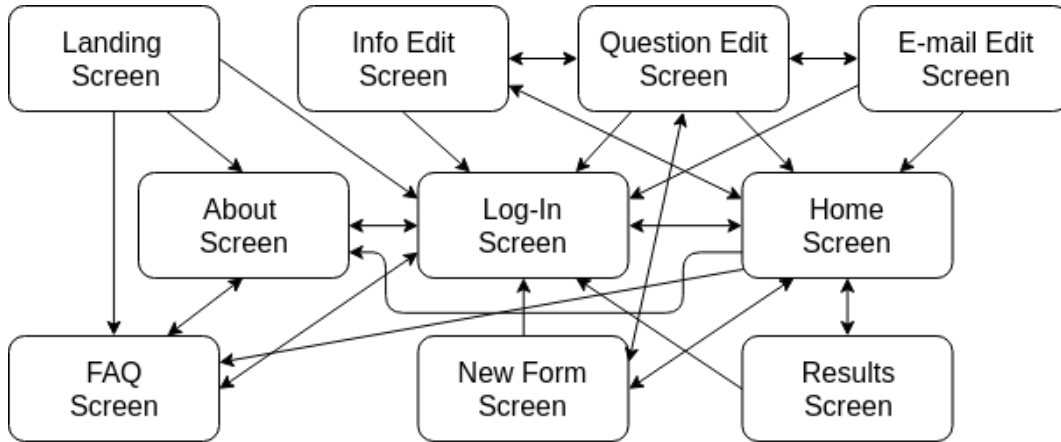


For ease of use, the team made the designs simple for each screen. On the top-right corner, there is the team EVAL logo and the name of the program. The top-right corner includes navigation buttons, typically a a “Home” button, which returns the user to the home screen, and the “Log Out” button, which takes the user to the log-in screen. The “content” is located below the top elements and contains data entry fields, evaluation results, and the like. In sections that may be look complicated, a help pop-up will appear when the mouse is over the question mark icon. Most screens also have a scroll bar on the right side if the information on a page cannot all fit in the web browser.

### 3 User Interface Walkthrough

This section goes into more detail about the screens in the user interface and how an administrator or instructor navigates through them. Figure 2 shows the navigation diagram, which shows the paths that a user can take through the system's interface. The diagram also includes the names for all the screens in the UI.

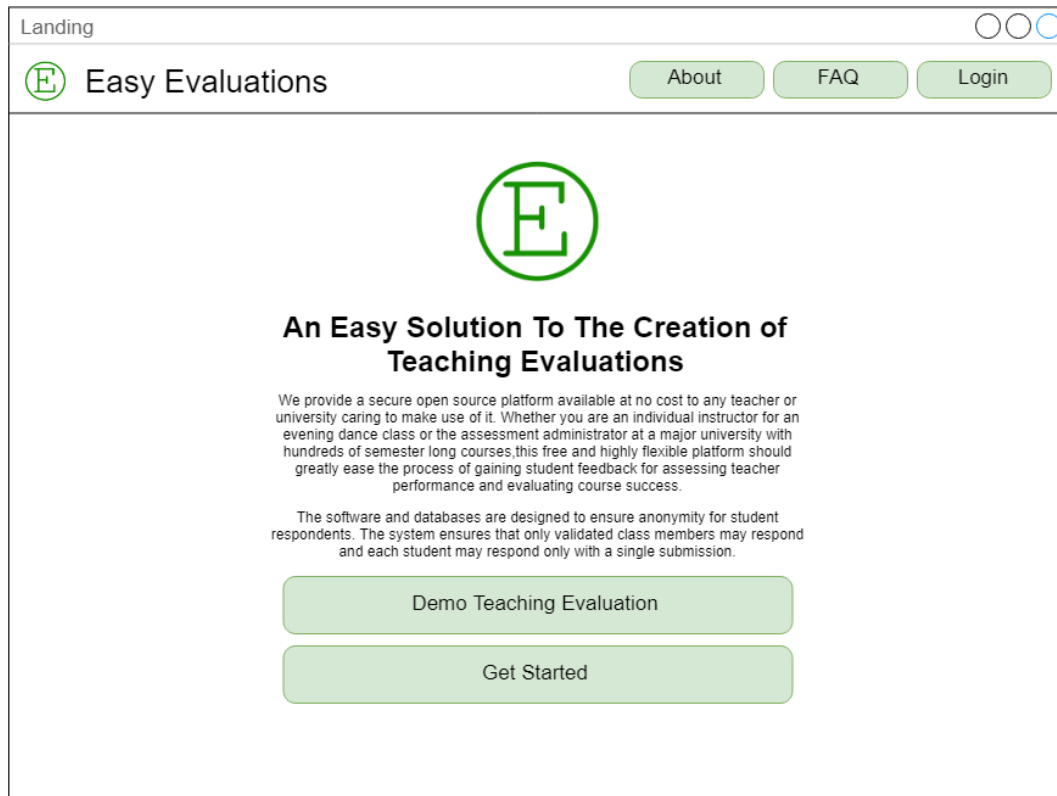
Figure 2: Navigation diagram of the user interface



The first screen that the user sees upon starting up the system is the **Landing Screen**. Next, the user can view the **About Screen**, **FAQ Screen**, or **Log-In Screen** and switch between the three. After entering a correct username and password on the log-in screen, the **Home Screen** appears. The user can then call up the **Info Edit Screen**, **New Form Screen**, or **Results Screen** from the home page or return to the about or FAQ screens. You must go to the new form screen to create a survey or the info edit screen to modify a survey. Next, the user proceeds to the **Question Edit Screen** and then the **E-mail Edit Screen**. The log-in screen and home screen can be accessed from any other screen except the landing, about, and FAQ screens.

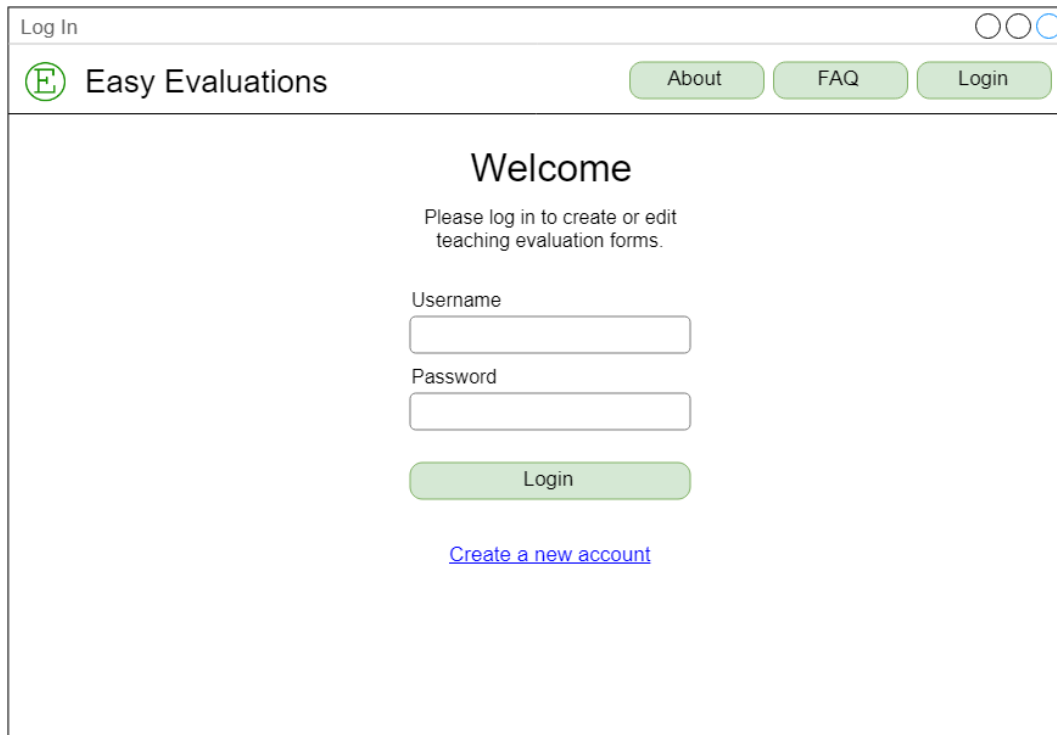
The next set of figures, Figures 3 to 12, consist of the wireframes for each screen in the evaluation system. These diagrams communicate the areas, menus, and buttons that are unique to a certain screen and what they do.

Figure 3: Landing screen



This is the first screen a user would see when entering the website. A user could click on the “Demo Teaching Evaluation” button, which would link to an informative video. Clicking “Get Started” would lead them to an account creation screen. Alternatively they could click “About”, “FAQ”, or “Sign In” to enter the corresponding screens.

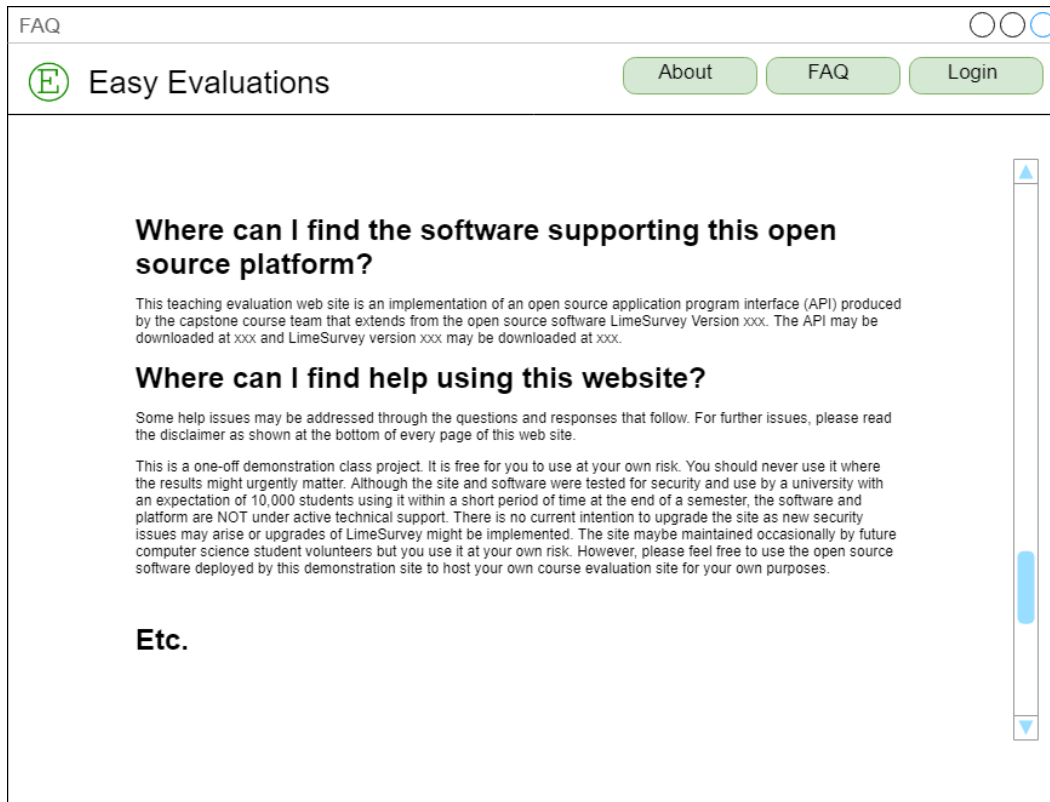
Figure 4: Log-in screen



The image shows a web browser window titled "Log In". The browser's address bar is empty. The page header features the "Easy Evaluations" logo on the left and three green buttons labeled "About", "FAQ", and "Login" on the right. The main content area has a "Welcome" heading, followed by the text "Please log in to create or edit teaching evaluation forms." Below this are two input fields: "Username" and "Password". A green "Login" button is positioned below the password field. At the bottom of the form area is a blue hyperlink that reads "Create a new account".

A user would enter a valid username and password to log in. Clicking “Log In” advances the user to the selection screen. The “About”, “FAQ”, and “Sign In” buttons would take you to their respective pages.

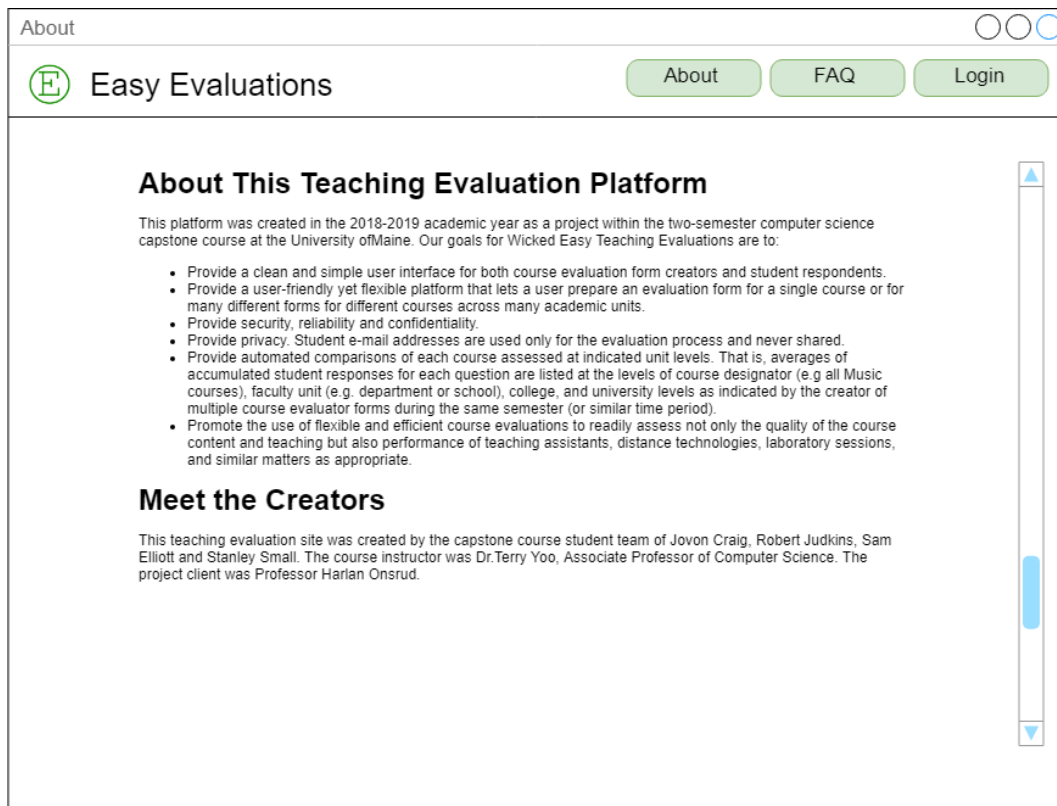
Figure 5: FAQ Screen



This screen details frequently asked questions that a user may have and their answers. The “About”, “FAQ”, and “Sign In” buttons would take you to their respective pages.



Figure 6: About screen



This screen gives details about the product and its purpose. It also credits the creators, as well as the client for whom the product was made. The “About”, “FAQ”, and “Sign In” buttons would take you to their respective pages.

Figure 7: Home screen

Home

**Easy Evaluations** [About](#) [FAQ](#) [Logout](#)

Choose one of the following actions:

- [Create a New Course & Teaching Evaluation Form](#)
- [Edit an Existing Unpublished Course Evaluation Form](#)

Select a Form...
  - COS 125 001
  - COS 140 001
  - COS 250 001
- [View Active Published Course Evaluations](#)
- [View Inactive Course Evaluations](#)
- [View Evaluation Results](#) [Download All Results](#)

Course Section	Course Designator	Instructor Last Name	Faculty Unit	College	University
Select a Report...					

This is the main screen a user would see after logging in. Users can choose to (1) create a new evaluation form, which will redirect them to the new form screen. They can (2) select a course which has been saved but unfinished and unpublished, redirecting them to the info edit screen. They can (3) select a course that has already been published but not completed, taking them to the edit screen where information is displayed but not editable. Also, they can (4) select a course that has been published and completed, taking them to the edit screen where information is displayed but not editable. Finally, they can (5) select to view evaluation results. One must choose a category type with the menu bar and a category name with the drop-down menu, making the results page appear.

Figure 8: New form screen

New Form

Easy Evaluations

Home Logout

### Course Information

If you want information on this form to be auto-filled from a previous submission that you can then edit, select a preset form below.

Select Preset ?

Course Designator ( e.g. MUS)

Course Number (e.g. 200)

Course Section (e.g. 001)

Course Title (e.g. Ballroom Dance)

Is this a graduate course? ☐ Yes ☐ No

Semester and Calendar Year (e.g. Fall 2019)

Save Next

This is the page users would see if they decided to create a new evaluation form. This page asks the user to fill out the general information of the course for which the evaluation form applies. Alternatively, the user can select a previously created preset rather than starting from scratch. See Appendix D page 1 for a full example of all fields that will be shown. The "Save" button will save the currently entered information. The "Next" button takes the user to the question edit screen.

Figure 9: Info edit screen

Edit Info Form

Easy Evaluations

Home Logout

### Edit Existing Unpublished Course Evaluation Form

{Course Identifier}

#### Course Information

Course Designator ( e.g. MUS)

Course Number (e.g. 200)

Course Section (e.g. 001)

Course Title (e.g. Ballroom Dance)

Is this a graduate course? ☐ Yes ☐ No

Semester and Calendar Year (e.g. Fall 2019)

Previous Page Save Next

Similar to the new form screen, a user would be redirected to this screen when choosing to edit an evaluation previously created but left unfinished or unpublished. The page allows users to view and edit the information of an existing unpublished evaluation form. The “Previous” button redirects users to the home page, the “Next” button redirects them to the question edit screen, and the “Save” button saves the entered information. See Appendix D for a full example of available questions to choose from.

Figure 10: Question edit screen

Edit Question Form

Easy Evaluations Home Logout

### Edit Existing Unpublished Course Evaluation Form {Course Identifier}

#### The Instructor

"How prepared was the instructor for class?" 1: "often unprepared", 5: "well prepared"	<input type="checkbox"/> Include	<input type="checkbox"/> Mandatory
"How clearly were the objectives of the course presented?" 1: "unclear", 5: "very clear"	<input type="checkbox"/> Include	<input type="checkbox"/> Mandatory
"Overall, how would you rate this instructor?" 1: "often unprepared", 5: "well prepared"	<input type="checkbox"/> Include	<input type="checkbox"/> Mandatory
Enter question here 1: <input type="text"/> , 5: <input type="text"/>	<input type="checkbox"/> Include	<input type="checkbox"/> Mandatory

Add Question ?

---

#### The Course

Previous Page Save Next

This is the second page a user would see when creating a course evaluation. You are redirected here after pressing the next button from the new form page or info edit page. The “Previous” button redirects the user to the question edit screen, the “Next” button goes to the e-mail edit screen, and the “Save” button saves the entered data. The page lists several generic questions that may be asked in a evaluation form. It also asks if the instructor would like to include a certain question and whether it should be mandatory. Users can also add custom questions at each section. See Appendix D for a full example.

Figure 11: E-mail edit screen

Edit Email Form

Easy Evaluations Home Logout

### Edit Existing Unpublished Course Evaluation Form

{Course Identifier}

#### Class Roll

Please insert in the field below the first name, last name and email of each student in the class. The information for each student should appear in a separate row and be separated by commas. (e.g. Mary, Smith, marysmith@gmail.com) You may cut and paste into the window at your option. Alternatively, upload a .csv file with the content in the first row of the file being first name, last name, e-mail.

Upload .csv ?

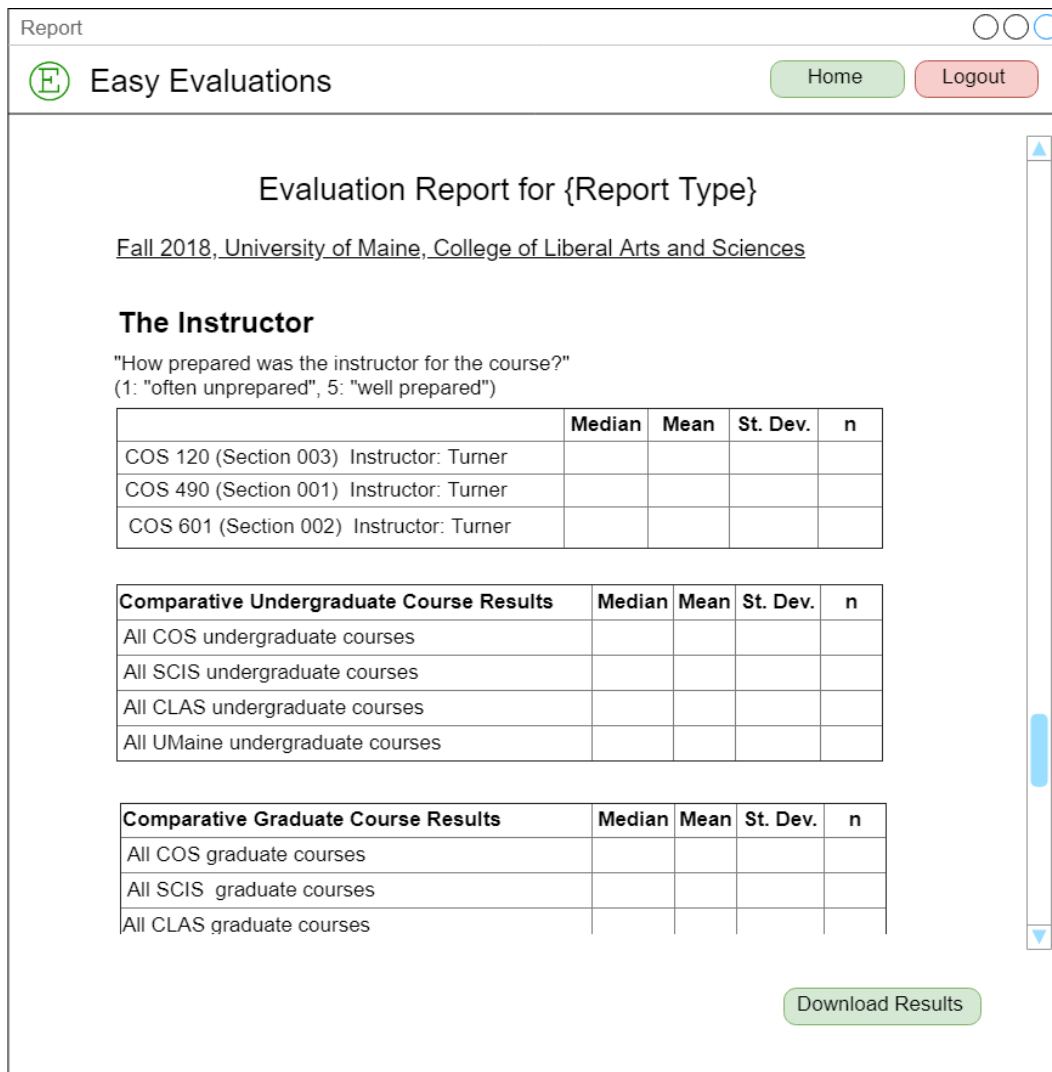
---

#### Emails to Be Sent to Students

Previous Page Save Publish ?

This is the final page a user would see when creating a new course evaluation. They will be redirected here when the “Next” button is pressed on the question edit Screen. It asks the instructor to include a list of students taking the course, to review a generic e-mail that would be sent to students, and to review a reminder e-mail that could be sent after a certain period of time. See Appendix D for a full example.

Figure 12: Results screen



This page is where a user would view results of their evaluations based on the category selected on the home screen. The responses are organized by each question in the evaluation and displays different tables for undergraduate courses and graduate courses. It includes the median, mean, standard deviation, and number of answers for each question along with question statistics for the classes in the selected category. The results can also be downloaded and exported as a .csv file by clicking the "Download Results" button. See Appendix D for a full example of an instructor report.

## 4 Data Validation

Table 1 lists the data items in the user interface of the evaluation system. A data item is an input that a user enters into the system and has a specified format. Many of the items listed are for editing a course evaluation form.

Table 1: Data item specification

Label	Screen(s)	Data Type	Format	Limit(s)
Username	Log in	String	username@gmail.com	N/A
Password	Log in	String	N/A	N/A
Course designator	New form, info edit	String	N/A	50 characters long
Course number	New form, info edit	String	N/A	50 characters long
Course section	New form, info edit	String	N/A	50 characters long
Course title	New form, info edit	String	N/A	50 characters long
Graduate course	New form, info edit	Boolean	N/A	50 characters long
Semester and year	New form, info edit	String	N/A	50 characters long
Faculty unit	New form, info edit	String	N/A	50 characters long
Department	New form, info edit	String	N/A	50 characters long
University	New form, info edit	String	N/A	50 characters long
Instructor first name	New form, info edit	String	N/A	50 characters long
Instructor last name	New form, info edit	String	N/A	50 characters long
Instructor e-mail	New form, info edit	String	N/A	50 characters long
Instructor phone	New form, info edit	String	N/A	50 characters long
Course evaluation administrator	New form, info edit	String	N/A	50 characters long
Evaluation administrator e-mail	New form, info edit	String	N/A	50 characters long
Starting assessment date	New form, info edit	String	mm/dd/yy	50 characters long
Mailing time	New form, info edit	String	hh:mm:ss	50 characters long
Closing assessment date	New form, info edit	String	N/A	50 characters long
Added questions	Question edit	Strings	N/A	150 characters long
1 score label	Question edit	String	N/A	50 characters long
5 score label	Question edit	String	N/A	50 characters long
“Include?” checkboxes	Question edit	Boolean	N/A	N/A
“Mandatory” checkboxes	Question edit	Boolean	N/A	N/A
Class roll	E-mail edit	String	Comma-separated lines: “first name, last name, e-mail address”	N/A
Initial e-mail to students	E-mail edit	String	N/A	N/A
Reminder e-mail	E-mail edit	String	N/A	N/A
Final confirmation e-mail	E-mail edit	String	N/A	N/A



## A Agreement Between Customer and Contractor

This page shows that all members of Team EVAL and the client, Harlan Onsrud, have agreed on all the information in the user interface design document. By signing this document, Team EVAL and Dr. Onsrud approve all of the designs for each screen in the interface, as well as how to navigate the interface.

The team will follow a process in the case that the design document is changed after we sign it. First, the team will write a rough draft of the changes to be made to the document. Second, all team members and Harlan Onsrud will sign the document agreeing to the changes. Finally, the team will make the changes to the final copy of the document.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
Jovon Craig	_____	_____
Sam Elliott	_____	_____
Robert Judkins	_____	_____
Stanley Small	_____	_____
Harlan Onsrud	_____	_____
Customer Comments:	_____	_____
_____		

# B Team Review Sign-off

This page shows that all members of Team EVAL have reviewed the user interface design document and agreed on its content. By signing this document, the team members agree that all information about the evaluation system's UI is accurate, and there is nothing in the document that is a source of contention.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
<b>Jovon Craig</b>	_____	_____
Comments:	_____	
_____		
<b>Sam Elliott</b>	_____	_____
Comments:	_____	
_____		
<b>Robert Judkins</b>	_____	_____
Comments:	_____	
_____		
<b>Stanley Small</b>	_____	_____
Comments:	_____	
_____		

## C Document Contributions

Stanley Small contributed to the discussion of the UI design and helped make revisions to the client's request. He also provided some formatting changes to the document. Stan contributed approximately 10 percent of the document.

Jovon Craig wrote the purpose of the document, the user interface standards section, the description of the UI navigation, the first three columns of the data item table, and Appendix C. He created the overall layout diagram, navigation diagram, and many of the individual screen layouts. He revised several screen layouts and parts of the document. Jovon contributed about 35 percent of the document.

Sam Elliott revised all of the individual screen layouts and their descriptions, and he added several new screens. He also wrote the last two columns in the data item table and revised the summaries of each wireframe. Sam contributed about 35 percent of the document.

Robert Judkins converted the UIDD template to the LaTeX format and placed it in our document. He initially wrote the descriptions under each wireframe picture of the UI. He also added the references and appendices A, B, and D. Robert contributed about 20 percent of the document.

## D Example Question Selection Form

### APPENDIX A SELECTION OF QUESTIONS AND CONTENT FOR INCLUSION ON A WEB-BASED STUDENT EVALUATION OF TEACHING FORM

#### INSTRUCTOR OR ACADEMIC UNIT SELECTION OF QUESTIONS FOR USE IN A WEB-BASED FORM:

If you want information on this form auto-filled from a previous submission that you may then edit, please **SELECT** (Note to Team: **SELECT** leads to a pulldown menu that includes all previous submission forms. When selected, all information would be copied from a previous form to the current form below except as noted.)

Course Designator (e.g. MUS)	_____
Course Number (e.g. 200)	_____
Course Section (e.g. 001)	_____
Course Title (e.g. Ballroom Dance)	_____
Is this a graduate course?	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> (bull's eye choices)
Semester and Calendar Year (e.g. Fall 2019)	_____ (don't auto populate)_
Name of Faculty Unit (e.g. School of Music)	_____
Name of College (e.g. Liberal Arts)	_____
Name of University (e.g. Univ of Maine)	_____
Last Name of Instructor (e.g. Smith)	_____ (need later in reporting to alphabetize by last name)
First Name of Instructor (e.g. Elizabeth)	_____
Instructor Email	_____
Instructor Phone	_____
Full Name of Course Evaluation Administrator	_____
Email of Course Evaluation Administrator	_____
Beginning Date of Assessments	_____ (use calendar pulldown) (don't auto populate)_
Time of Day for Any and All Mailings	_____ (use time select)_____ (don't auto populate)_
Closing Date	_____ (use calendar pulldown) (don't auto populate)_

#### NOTE TO CAPSTONE TEAM:

Variable names from some of the entries above include: {COURSEDESIGNATOR}, {COURSENUMBER}, {COURSETITLE}, {ADMINNAME}, {ADMINEMAIL}, {BEGINNINGDATE}, {CLOSINGDATE}, {TIME}. Some other variable names likely to be used coming from the student roll upload include: {FIRSTNAME}, {LASTNAME}, {EMAIL}.

# UNIVERSITY OF MAINE STUDENT EVALUATION OF TEACHING

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>THE INSTRUCTOR</b>			
How prepared was the instructor for class?	often unprepared; well prepared		
How clearly were the objectives of the course presented?	unclear; very clear		
How enthusiastic was the instructor about the subject?	very little; very much		
How clearly did the instructor present concepts, principles and theories?	unclear; very clear		
How much were you encouraged to think for yourselves?	very little; very much		
How concerned was the instructor for the quality of student learning?	unconcerned; very concerned		
Did the instructor show respect for the questions and opinions of the students?	rarely; always		
Did the instructor ensure an environment of respect for all groups of people in the classroom?	rarely; always		
Did the instructor inspire confidence in his/her knowledge?	very little; very much		
<b>Overall, how would you rate the instructor?</b>	poor; excellent		
Please list further questions to be added regarding the instructor, if any, indicating a 1-5 rating scale (i.e. 5 is best)			
(a)			
(b) Online creator should be able to add up to 15 questions			
<b>THE COURSE</b>			
Were class meetings profitable and worth attending?	rarely; always		
What is your overall rating of the primary readings?	poor; excellent		
How much did this course challenge you intellectually?	very little; very much		
How much did you learn from this course?	very little; very much		
<b>What is your overall rating of this course?</b>	poor; excellent		
Please list further questions to be added regarding the course, if any, indicating a 1-5 rating scale (i.e. 5 is best)			
(a)			
(b) Online creator should be able to add up to 15 questions			
<b>STUDENT WORK ASSESSMENT</b>			
Did the instructor let you know what was expected on the assessments (exams, assignments, projects, papers, etc.)?	unclear; very clear		
Did the assessments reflect the important aspects of the course?	rarely; always		
How fair were the grading procedures?	unfair; completely		
Overall, how would you rate the assessment process (exams, assignments, projects, papers, etc.)?	poor; excellent		
Please list further questions to be added regarding the assessment, if any, indicating a 1-5 rating scale (i.e. 5 is best)			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **laboratory questions** were copied from a previous University of Maine approved longer bubble form.

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>THE LABORATORY EXPERIENCE</b>			
Did this course have one or more regularly scheduled laboratory sessions? (If the student answer is yes, the following questions appear)			
How much did the laboratory experience contribute to your learning in this course?	very little, very much		
<b>Overall, how would you rate the laboratory experience?</b>	poor; excellent		
What was done particularly well in the laboratory experience?	----		
How could the laboratory experience be improved?	----		
Please list further lab questions to be added, if any, and a rating scale			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **teaching assistant questions** were copied from a previously used UMaine longer bubble form.

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>THE TEACHING ASSISTANT</b>			
Was there a teaching assistant supporting this course? (If answer is yes, the following questions appear)			
If there was more than one teaching assistant for the course, please name the TA you are evaluating. (open ended response)			
How much did the teaching assistant contribute to your learning in this course?	very little, very much		
How concerned was the teaching assistant for the quality of student learning?	unconcerned; very concerned		
Did the teaching assistant show respect for the questions and opinions of students?	rarely; always		
Did the teaching assistant inspire confidence in his/her knowledge?	very little; very much		
Would you want to have this teaching assistant in the future in another course?	definitely not, definitely yes		
Would you recommend this teaching assistant to assist in this course in the future?	definitely not, definitely yes		
<b>Overall, how would you rate the teaching assistant?</b>	poor; excellent		
Name something the teaching assistant did particularly well.	----		
Name something the teaching assistant could do better in the future.	----		
Please list further TA questions to be added, if any			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **online component questions** were copied from the current UMaine Augusta distance education form

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>ONLINE COMPONENT ASSESSMENT</b>			
Did you take this course as a distance learning student? (If answer is yes, the following questions appear)			
Please indicate the primary online modality used with the course. [only one selection allowed] <radio> Sessions were primarily <u>synchronous</u> in that I was typically required or highly encouraged to interact live online with the instructor and other students in most or all sessions. <radio> Sessions were primarily <u>asynchronous</u> in that I primarily viewed recorded video lectures or class sessions on my own schedule and had only one or fewer opportunities per week to interact live online with the instructor or other students. <radio> Sessions were approximately an equal mix of <u>synchronous</u> and <u>asynchronous</u> interactions. <radio> Other. Describe: <short text box>	----		
The online modality used with the course was well suited to my needs.	strongly disagree, strongly agree		
There was adequate opportunity for me to interact with the instructor.	strongly disagree, strongly agree		
There was adequate opportunity for me to interact with other students.	strongly disagree, strongly agree		
The online technologies used in this course worked the way they were supposed to.	strongly disagree, strongly agree		
The communication tools were easy to use (email, assignment delivery, exam delivery or proctoring, chat, blog, teleconferencing, online courseware, web, etc.).	strongly disagree, strongly agree		
Technology support was there if I needed it.	strongly disagree, strongly agree		
The online experience was well-suited to the way I like to learn.	strongly disagree, strongly agree		
Which statement best characterizes your belief after having taken this course? [only one selection allowed] <radio> I learned MUCH LESS in this online class than I probably would have in the traditional format. <radio> I learned A LITTLE LESS in this online class than I probably would have in the traditional format. <radio> I learned ABOUT THE SAME in this online class than I probably would have in the traditional format. <radio> I learned A LITTLE MORE in this online class than I probably would have in the traditional format. <radio> I learned MUCH MORE in this online class than I probably would have in the traditional format.	----		

Which statement best characterizes your belief about the grade you expect to receive in this course? [only one selection allowed]	----		
<radio> I worked MUCH LESS for my grade in this online class than I probably would have had to in the traditional format.			
<radio> I worked A LITTLE LESS for my grade in this online class than I probably would have had to in the traditional format.			
<radio> I worked ABOUT THE SAME for my grade in this online class than I probably would have had to in the traditional format.			
<radio> I worked A LITTLE HARDER for my grade in this online class than I probably would have had to in the traditional format.			
<radio> I worked MUCH HARDER for my grade in this online class than I probably would have had to in the traditional format.			
Please identify an e-learning aspect of the course that you found particularly valuable or beneficial	----		
Please identify an e-learning aspect of the course that could be improved.	----		
Please list further questions to be added, if any			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **open-ended questions** were copied from the current University of Maine bubble form.

Questions	1-5 Rating Scale	Include question? (If yes, indicate with ✓)	Make response mandatory? (If yes, indicate with ✓)
<b>OPEN ENDED QUESTIONS</b>			
Please identify the aspects of this course that were of most value to you.	----		
Please mention at least one additional topic or component that you would like to see included in this course.	----		
Please make any additional comments that you desire to make about the course instructor, materials or pedagogy.	----		
Please list further open-ended questions to be added, if any			
(a)			
(b) Online creator should be able to add up to 15 questions			

#### Additional Option for Mandatory Questions

For questions in the tables above that you have designated as mandatory, should the last response option for possible selection on all of these questions be “decline to respond? (If “yes”, an option 6 will be added) Yes \_\_\_ No \_\_\_

#### CLASS ROLL

**Please insert in the field below the first name, last name and email of each student in the class.** The information for each student should appear in a separate row and be separated by commas. (e.g. Mary, Smith, marysmith@gmail.com) You may cut and paste into the window at your option.

<window for entry – do not autopopulate in future subsequent uses of the form>

Alternatively, upload a cvs file with the content in the first row of the file being firstname, lastname, email. **UPLOAD**

#### EMAILS TO BE SENT TO STUDENTS

**Initial Email invitation to Participate**



Although we recommend that you do not change the following email text, you may edit the *Invitation to Participate* as appropriate for your purposes if needed. Do NOT change any item listed as a {VARIABLE} because this will cause an error in your submission that you will be forced to correct prior to successful submission. This email will be sent to each student on the begin date at the time you specified above.

<window for text – Team, Please insert from below the suggested text for DEFAULT INITIAL INVITATION TO STUDENTS>

**Do you want one or more reminder emails sent to students who have yet to respond after a few days?** Yes \_\_\_ No \_\_\_  
(If the answer is yes, the following statements and questions appear)

#### **Reminder Emails**

You may send up to three reminder emails to those students who have yet to complete the teaching evaluation. The text for all reminder emails will be identical.

Although we recommend that you do not change the following email text, you may edit the *Reminder Emails* to students as appropriate for your purposes if needed. Do NOT change any item listed as a {VARIABLE} because this will cause an error in your submission that you will be forced to correct prior to successful submission.

<window for text – Team, Please insert from below the suggested text for DEFAULT REMINDER TO STUDENTS>

Reminder emails are sent spaced three days apart until the closing date is reached unless you specify a longer interval.

Please send reminder emails at intervals of <insert number of 3 or above> days.

Only a maximum of three reminders will be sent regardless of the spacing. Reminder mailings are sent at the same time of day as the initial invitation mailing.

#### **Confirmation Message**

Although we recommend that you do not change the following email text, you may edit the *Confirmation Message* reminder to students as appropriate for your purposes if needed. Do NOT change any item listed as a {VARIABLE} because this will cause an error in your submission that you will be forced to correct prior to successful submission.

<window for text – Team, Please insert from below the suggested text for DEFAULT CONFIRMATION>

**Please double check all information provided above. Once you click SUBMIT you may not make further changes unless you receive an error notification. You will receive a summary of the student responses by email at the time and date you designated for the teaching evaluation process to end.**

IMPORTANT: You have NOT successfully completed your assessment submission until you press the SUBMIT button below AND you are taken to a web page that states your submission has been successful. If you are returned to an earlier portion of this form, complete all information items now highlighted in red and then click SUBMIT again.

#### **SUBMIT**

+++++

**DEFAULT INITIAL INVITATION TO STUDENTS FOR EDITING BY THE  
INSTRUCTOR/ADMINISTRATOR USER**

+++++

Subject: Invitation to Complete Evaluation for {COURSEDESIGNATOR} {COURSENUMBER} {COURSETITLE}

Dear {FIRSTNAME},

Please complete the teaching and course evaluation for {COURSEDESIGNATOR} {COURSENUMBER}  
{COURSETITLE}.

This student evaluation of teaching is completely anonymous unless you purposefully identify yourself in response to one of the questions. The software system will send you automatic reminders every few days until you complete the evaluation.

To respond, simply click the link at the end of this message.

Sincerely,  
{ADMINNAME}

-----  
Click here to complete the teaching and course evaluation:  
{SURVEYURL}

{ADMINNAME} ({ADMINEMAIL})

++++++  
**DEFAULT REMINDER TO STUDENTS FOR EDITING BY THE INSTRUCTOR/ADMINISTRATOR USER**  
– TO BE SENT AT REGULAR INTERVAL UNTIL STUDENT RESPONDS, MAXIMUM OF THREE IS REACHED,  
OR CLOSING DATE IS REACHED  
++++++

Subject: Reminder to Complete Evaluation for {COURSEDESIGNATOR} {COURSENUMBER} {COURSETITLE}

Dear {FIRSTNAME},

Recently we invited you to complete a teaching evaluation for “{COURSEDESIGNATOR} {COURSENUMBER} {COURSETITLE}”. We note that you have not yet completed the evaluation, and wish to remind you that it is still available should you wish to take part.

To participate, please click on the link below.

The deadline for completing the evaluation is {CLOSINGDATE} at {TIME}.

Your participation is extremely important to the improvement of teaching and courses at the {UNIVERSITYNAME}

Sincerely,  
{ADMINNAME}

-----  
Click here to complete the teaching and course evaluation:  
{SURVEYURL}

**NOTE TO TEAM: SHOULD THE FOLLOWING GREEN OPTION BE INCLUDED AND IMPLEMENTED?**  
If you do not want to participate in the student evaluation process and don't want to receive any more email reminders, please click the following link:  
{OPTOUTURL}

Note: Such an option is required in many instances by “Do Not Spam” federal legislation but would not be required typically in a university mailing to your own students. Implement as good practice regardless?

{ADMINNAME} ({ADMINEMAIL})

++++++  
**DEFAULT CONFIRMATION**  
++++++

Subject: Completion of Course Evaluation for {COURSENUMBER} {COURSETITLE}

Dear {FIRSTNAME},

This email is to confirm that you have completed a student evaluation of {COURSEDESIGNATOR}  
{COURSENUMBER} {COURSETITLE} at the {UNIVERSITYNAME}. Your response has been saved. Thank you for  
participating.

If you have any further questions about this email, please contact {ADMINNAME} at {ADMINEMAIL}.

Sincerely,

{ADMINNAME}

## E Example Results Display

### REPORT FOR INSTRUCTOR: ROY TURNER

(includes all courses taught by this instructor this semester under this administrator account)

Fall 2018
University of Maine (UMaine)
College of Liberal Arts and Sciences (CLAS)

#### THE INSTRUCTOR

How prepared was the instructor for class? (1- often unprepared; 5- well prepared)				
	Median	Mean	Std. Dev	n
COS 120 (Sec 003) Instr: Turner				
COS 490 (Sec 001) Instr: Turner				
INT 601 (Section 002) Instr: Turner				
COMPARATIVE UNDERGRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All COS undergraduate courses (submitted under this admin account for this semester)				
All SCIS undergraduate courses (Faculty Unit - this admin account for this semester)				
All CLAS undergraduate courses (College - this admin account for this semester)				
All UMaine undergraduate courses (University - this admin account for this semester)				
COMPARATIVE GRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All INT graduate courses (submitted under this admin account for this semester)				
All SCIS graduate courses (Faculty Unit - this admin account for this semester)				
All CLAS graduate courses (College - this admin account for this semester)				
All UMaine graduate courses (University - this admin account for this semester)				

How clearly were the objectives of the course presented? (1- unclear; 5- very clear)				
	Median	Mean	Std. Dev	n
COS 120 (Sec 003) Instr: Turner				
COS 490 (Sec 001) Instr: Turner				
INT 601 (Section 002) Instr: Turner				
COMPARATIVE UNDERGRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All COS undergraduate courses (submitted under this admin account for this semester)				
All SCIS undergraduate courses (Faculty Unit - this admin account for this semester)				
All CLAS undergraduate courses (College - this admin account for this semester)				
All UMaine undergraduate courses (University - this admin account for this semester)				
COMPARATIVE GRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All INT graduate courses (submitted under this admin account for this semester)				
All SCIS graduate courses (Faculty Unit - this admin account for this semester)				
All CLAS graduate courses (College - this admin account for this semester)				
All UMaine graduate courses (University - this admin account for this semester)				

Etc.

Overall, how would you rate the instructor? (1- poor; 5- excellent)				
	Median	Mean	Std. Dev	n
COS 120 (Sec 003) Instr: Turner				
COS 490 (Sec 001) Instr: Turner				
INT 601 (Section 002) Instr: Turner				
COMPARATIVE UNDERGRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All COS undergraduate courses (submitted under this admin account for this semester)				
All SCIS undergraduate courses (Faculty Unit - this admin account for this semester)				
All CLAS undergraduate courses (College - this admin account for this semester)				
All UMaine undergraduate courses (University - this admin account for this semester)				
	Median	Mean	Std. Dev	n

COMPARATIVE GRADUATE COURSE RESULTS				
All INT graduate courses (submitted under this admin account for this semester)				
All SCIS graduate courses (Faculty Unit - this admin account for this semester)				
All CLAS graduate courses (College - this admin account for this semester)				
All UMaine graduate courses (University - this admin account for this semester)				

Note that the following is a **Further Question** and thus the question is **not** compared with other grouped evaluations

Is this instructor well groomed? (1- never; 5- always)				
	Median	Mean	Std. Dev	n
COS 120 (Sec 003) Instr: Turner				
COS 490 (Sec 001) Instr: Turner				
INT 601 (Section 002) Instr: Turner				

Etc.

#### THE COURSE

Were class meetings profitable and worth attending? (1- rarely; 5- always)				
	Median	Mean	Std. Dev	n
COS 120 (Sec 003) Instr: Turner				
COS 490 (Sec 001) Instr: Turner				
INT 601 (Section 002) Instr: Turner				
COMPARATIVE UNDERGRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All COS undergraduate courses (submitted under this admin account for this semester)				
All SCIS undergraduate courses (Faculty Unit - this admin account for this semester)				
All CLAS undergraduate courses (College - this admin account for this semester)				
All UMaine undergraduate courses (University - this admin account for this semester)				
	Median	Mean	Std. Dev	n
COMPARATIVE GRADUATE COURSE RESULTS				
All INT graduate courses (submitted under this admin account for this semester)				

All SCIS graduate courses (Faculty Unit - this admin account for this semester)				
All CLAS graduate courses (College - this admin account for this semester)				
All UMaine graduate courses (University - this admin account for this semester)				

Etc.

#### OPEN ENDED QUESTIONS

Please identify the aspects of this course that were of most value to you.	
	Volunteered Self Identification by a Student and Verified
<b>COS 120 (Sec 003) Student Comments</b>	
The jokes.	Anonymous
Instructor really knew the material.	Jovon Adams
The exercises were great.	Anonymous
<b>COS 490 (Sec 001) Student Comments</b>	
The readings were marvelous	Anonymous
It provided a great hour for napping	Anonymous
The insights of my student peers.	Sam Sneed
<b>INT 601 (Section 002) Student Comments</b>	
This course sucked.	Anonymous
The snacks were good.	Betty Snow
The case studies were the most important part of the course.	Anonymous
The homework exercises were great.	Anonymous

Etc.

## E Code Inspection Report

# Code Inspection Report

Course Evaluation System

Team EVAL

Jovon Craig, Sam Elliott, Yuanqi Guo, Robert Judkins, and Stanley Small

Client: Dr. Harlan Onsrud

March 14, 2019

University of Maine - Spring 2019 - COS 497  
Instructor: Professor Terry Yoo





Course Evaluation System  
Code Inspection Report

**Contents**

# 1 Introduction

Team EVAL is creating a system to more efficiently create and distribute post-semester teaching evaluations. Our client wants us to build such a system because the current one used by the University of Maine is too difficult to use. To ensure that we are on the right direction, our system must be properly inspected to be free from defects and meet all the functional requirements as laid out in the System Requirements Specification.

## 1.1 Purpose of This Document

The code inspection report is a detailed overview of the walkthrough our team made to ensure that the course evaluation system works as intended. The first section of the document mentions the coding conventions that the team used as guidelines, as well as a list of defects we checked during the review. The second section details our inspection process, how it deviated from a typical inspection, and Team EVAL's reflections. The last sections describe the system components inspected and the defects we found.

This document is intended for the development team, the product client, Dr. Harlan Onsrud, and potential users of the system. Team EVAL needs this document to fix the defects that we found during inspection and keep the defect checklist in mind. Dr. Onsrud needs it to confirm that Team EVAL is meeting requirements and caring about code quality. Lastly, the document helps the software's users by telling how the actual modules differ from how they were designed in the SDD.

## 1.2 References

Craig, J., Elliott, S., Judkins, R., & Small, S. 29 October 2018. *System Requirements Specification*.

Craig, J., Elliott, S., Judkins, R., & Small, S. 16 November 2018. *System Design Document*.

Van Rossum, G., Warsaw, B., & Coghlan, N. (2001, July 5). *PEP 8 – Style Guide for Python Code*. Retrieved from <https://www.python.org/dev/peps/pep-0008/>

Goodger, D., & Van Rossum, G. (2001, May 29). *PEP 257 – Docstring Conventions*. Retrieved from <https://www.python.org/dev/peps/pep-0257/>

*Google JavaScript Style Guide*. (n.d.). Retrieved March 4, 2019, from <https://google.github.io/styleguide/jsguide.html>

## 1.3 Coding and Commenting Conventions

Team EVAL has adopted several guidelines for proper code and commenting to have a more consistent code base. In code inspections these standards will be referenced better understand what kinds of defects may be present in the code inspected.

For the Python code in the back-end API, the team is following the official Python style guide, PEP 8, and the docstring conventions in PEP 257. Available on the Python website, PEP 8 and 257 were written in 2001 by the creator of Python, Guido van Rossum, and other core developers. Both guides in turn were derived by van Rossum's original style guide for Python. PEP 8 covers topics such as indentation, line length, imports, whitespace, comments, and naming. PEP 257 has brief guidelines on documentation strings.

For the JavaScript code written on the front end, Team EVAL is following the Google JavaScript Style Guide. Google uses JavaScript so much for its front-end projects that it wrote conventions on how to properly write in the language. The standards in this guide are followed by Google's developers for their own development. The style guide includes guidelines on issues like source file structure, indentation, whitespace, comments, literals, naming, and JavaScript's documentation language, JSDoc.

## 1.4 Defect Checklist

The team created a list of possible defects that could be encountered during our inspection. Much of the defect list references the style guides mentioned in the previous section. Each defect falls under one of four

categories: violations of coding conventions, control flow errors, flaws in the system’s security, and violations of commenting standards. The list is shown in Table 1 on the next page:

Table 1: Defect checklist

Category	Statement
Coding Style	Indentation is done with tabs instead of spaces
Coding Style	Indentation amount hurts code clarity
Coding Style	Multiple imports on the same line
Coding Style	Extraneous whitespace around punctuation
Coding Style	Unnecessary blank lines
Coding Style	Unnecessary compound statement
Coding Style	Too many units in a single script file
Coding Style	Return types in same function are inconsistent
Coding Style	Double quotes in string literal (JavaScript)
Coding Style	No braces in control structure (JavaScript)
Coding Style	JavaScript name is not in lower camel case
Coding Style	Python name is not in lowercase with underscores
Coding Style	Constant name is not in all-caps
Commenting Style	Comment is unnecessary for understanding
Commenting Style	Function or class has no docstring
Commenting Style	Function docstring does not mention parameter and return types
Commenting Style	Multi-line docstring has no summary line
Control Flow	Off-by-one error in a loop
Control Flow	Non-explicit exception handling
Control Flow	Too many paths in a control structure
Security	Default password used for authentication
Security	No authentication in code requiring security
Security	Unsanitized user input
Security	Open TCP/IP ports
Other	Function has no unit test(s)

## 2 Code Inspection Process

The code inspection was conducted with all members of Team EVAL present in a single sitting. The code was reviewed beforehand by each member, and a preliminary check was completed by each author. The following sections document the findings of our inspection.

### 2.1 Description

The code inspection process that Team EVAL followed was more informal than a typical Fagan inspection. Our team compared the process that we followed at the inspection meeting to a walkthrough as described by Williams in the text Intro to Software Engineering. Our process diverged from the typical inspection in an attempt to save time, allow for a more flexible meeting style, and a more casual atmosphere.

Before the inspection, the team decided the time and location of the meeting and who would take each role. Next, Stan and Jovon wrote the defect checklist so that the team could use it as a reference for what defects could be found. During the inspection, Stan led the proceedings as the “moderator”, and Sam was the “recorder”, typing down any defects encountered. The author of a module took on the “author” and “reader” roles. All members of the team were “inspectors”, pointing out flaws in the code. Some flaws referenced the style guides described earlier in this report.

## 2.2 Impressions of the Process

Our inspection process proved to be very effective. We were able to find a number of defects or simple fixes that may not have been caught without the inspection. It also helped the team focus on continuity and consistency in our code. It was very beneficial to sit down as a team and walk through every file in our growing project. As the project grows in size, it is easy for defects to fall through the cracks when trying to produce high-quality code. Sitting down as a team and systematically going through every file in our repository caused us to double-check work that we previously may not have reviewed. Since our project has several separate components, it was useful for each of us to see files that we may have written and to have the original authors explain them.

We were able to find defects and discuss possible solutions to them quickly. We did not waste time delving into exact solutions of larger issues, but many of our defects were quick fixes and it was important to discuss them as a team. The process seemed beneficial enough for a team and project of this size that it would be useful to do the exact same process several times throughout development.

The best modules as determined by the team were contained in the back-end files. While some of these files were generated by tools and frameworks, the code clearly followed the coding standards outlined in this document and contained more than sufficient documentation. Specific files which showed great promise include `teachers.controller.py` and `test_teachers.controller.py`.

Many of the worst modules, specifically `QuestionsForm.js` and `EnrollForm.js`, were contained in the front end made with the React framework. The code showed a distinct lack of understanding of the framework and deviated quite strongly from the system design document. As a consequence, the team has decided to revisit these sections of the code to better reflect decisions made early in the design process.

## 2.3 Inspection Meetings

We held one inspection meeting to make our code inspection report. This meeting took place in a physical location, Boardman Hall 136. It occurred on March 6, 2019, from 3 PM to 4:15 PM. Everyone on Team EVAL (Jovon, Sam, Tom, Robert, and Stan) participated in the inspection. Stan was the moderator of the inspection, Sam was the recorder, whoever wrote a certain modular unit served as the author and reader, and all team members were inspectors. The meeting covered every code unit in the system that is not automatically generated, including testing files, JavaScript files, and Python scripts.

## 3 Modules Inspected

In our inspection meeting, we looked through any module that was integral to our course evaluation system. The team defines a module as any individual script that the system can run. Table 2 on the next page lists each module examined, its functionality, and how it fits into the design mentioned in the SDD. Note that a few modules have been replaced by new ones since our inspection.

Table 2: List of modules inspected

Name	Functionality	Relation to System Design
.travis.yml	Defines which tests are used in continuous integration	For testing, not in system design document
docker-compose.yml	Defines the docker-compose command	For project building, not in system design document
About.js	What is displayed on the About page	Part of React front end
App.js	Old file which ran the application, replaced by AppRouter.js	Part of React front end
AppRouter.js	Routes the application to different pages, contains the current state of the app as a whole	Part of React front end
CourseForm.js	Page on which users define a new evaluation form	Part of React front end
EditCourse.js	Page on which users edit old courses, replaced by CourseForm.js	Part of React front end
EnrollForm.js	Page on which users define who takes the evaluation form	Part of React front end
FAQ.js	What is displayed on the FAQ page	Part of React front end
Home.js	What is displayed on the Home page	Part of React front end
Landing.js	What is displayed on the Landing page	Part of React front end
Login.js	What is displayed on the Login page, allows users to log in	Part of React front end
QuestionForm.js	Page on which users enter which questions they want in the survey	Part of React front end
Index.js	Used in construction of JS app	Part of React front end
insert_mock_data.sql	Inserts mock data into the database	For testing, not in system design document
test_teachers_controller.py	Runs tests for back-end API	For testing, not in system design document
teachers_controller.py	Processes API calls	Part of Python API
limesurvey.py	Interfaces with LimeSurvey	Part of Python back end

Some of the modules in the system are not yet completed. User login has not yet been completed, but will be finished after spring break. Additionally, some pages originally promised to the client have not been implemented. The team is focused on the most essential elements of the application, and has notified the client accordingly.

There are several differences between the actual design of a few of the modules and what was proposed in the System Design Document. The front end is supposed to exploit the features of the React library for user interaction, but Sam and Robert instead used mostly HTML for this. They are currently reworking the front end so that React.js is used properly.

In the back end, the endpoint “GET publish” was added so that the API could create a .txt file of a survey and publish it on LimeSurvey. The “POST results” endpoint was removed because there seemed to be no need for it. Also, the API now retrieves a JSON object of survey results instead of a .txt file. The team did not mention using an external library (“limesurvey.py”) to interface with LimeSurvey because we did not know it would be helpful.

## 4 Defects

During the inspection, the team found multiple defects in the system, spread across many different components. Table 3 on the next page gives a list of all the defects that we found. Each defects falls under one of five categories: correctness to the functional requirements, coding conventions, commenting conventions, user friendliness, security, and miscellaneous flaws.

Table 3: List of defects found

Module	Category	Description
.travis.yml	Other	Lacks back-end API tests
.travis.yml	Commenting	Does not have enough comments
docker-compose.yml	Commenting	Lacks comments
Front-end modules	Commenting	No header comment in each file
Front-end modules	Commenting	No docstrings
Front-end modules	Coding Convention	Double quotes used in some strings
Front-end modules	Correctness	No input validation
Front-end modules	User Friendliness	No error messages
Front-end modules	Security	Log-in validation ignored after authentication token generated
Front-end modules	Other	Files are not in subdirectories
Front-end modules	Other	No tests
Front-end modules	User Friendliness	CSS buttons too big
About.js	Commenting	No comments
About.js	User Friendliness	Displayed text is outdated
Form.js	Coding Convention	Some names not in lower camel case
Form.js	Correctness	No global variable for API
EditCourse.js	Other	Redundant file
EnrollForm.js	Coding Convention	Has too little whitespace
QuestionForms.js	Coding Convention	Poor formatting
QuestionForms.js	Other	Too long to be one script
Results.js	Correctness	Displays mock data
Back-end modules	Other	Has unnecessary “admins” scripts
insert_mock_data.sql	Correctness	Does not insert all required survey tags
test_teachers.controller.py	Commenting	Lacks comments
teachers_controller.py	Coding Convention	Has multiple imports in one line
teachers_controller.py	Security	Uses default MySQL username and password
teachers_controller.py	Security	MySQL password exposed in repo
teachers_controller.py	Commenting	Too few comments
teachers_controller.py	Coding Convention	Some lines are too long
teachers_controller.py	Coding Convention	Inconsistent string formatting methods
limesurvey.py	Commenting	Lacks comments

## A Agreement Between Customer and Contractor

This page shows that all members of Team EVAL and the client, Harlan Onsrud, have agreed on all the information in the code inspection report. By signing this document, Team EVAL and Dr. Onsrud approve the coding conventions used, the information about the inspection process, and the descriptions of the defects and inspected modules.

The team will follow a process in the case that the document is changed after we sign it. First, the team will write a rough draft of the changes to be made to the document. Second, all team members and Harlan Onsrud will sign the document agreeing to the changes. Finally, the team will make the changes to the final copy of the document.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
Jovon Craig	_____	_____
Sam Elliott	_____	_____
Yuanqi Guo	_____	_____
Robert Judkins	_____	_____
Stanley Small	_____	_____
Dr. Harlan Onsrud	_____	_____
Customer Comments:	_____	_____
_____		

# B Team Review Sign-off

This page shows that all members of Team EVAL have reviewed the code inspection report and agreed on its content. By signing this document, the team members agree that all information about Team EVAL’s code inspection is accurate, and there is nothing in the document that is a source of contention.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
<b>Jovon Craig</b>	_____	_____
Comments:	_____	
_____		
<b>Sam Elliott</b>	_____	_____
Comments:	_____	
_____		
<b>Yuanqi Guo</b>	_____	_____
Comments:	_____	
_____		
<b>Robert Judkins</b>	_____	_____
Comments:	_____	
_____		
<b>Stanley Small</b>	_____	_____
Comments:	_____	
_____		



## C Document Contributions

Stanley Small discussed the status of various modules and helped to describe the outcomes of the inspection. He proofread the document for final submission. Stan contributed approximately 20 percent of the document.

Jovon Craig wrote the purpose of the document, references, coding conventions sections, part of the defect checklist, description of the inspection process, and list of defects. He also made several revisions to the document. Jovon contributed about 40 percent of the document.

Yuanqi Guo wrote more content for the introduction and revised the formatting in the document. Tom contributed about 5 percent of the document.

Sam Elliott added the table of the modules that the team inspected and made several small additions to the document. Sam contributed about 20 percent of the document.

Robert Judkins wrote the section stating the team's impressions of the inspection process. He contributed about 15 percent of the document.

## **F Administrator Manual**

# Administrator Manual

## Course Evaluation System

Team EVAL

Jovon Craig, Sam Elliott, Yuanqi Guo, Robert Judkins, and Stanley Small

Client: Dr. Harlan Onsrud

April 4, 2019

University of Maine - Spring 2019 - COS 497  
Instructor: Professor Terry Yoo



# Course Evaluation System

## Administrator Manual

### Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose of This Document . . . . .	2
1.2	References . . . . .	2
<b>2</b>	<b>System Overview</b>	<b>2</b>
2.1	Background . . . . .	2
2.2	Hardware and Software Requirements . . . . .	2
<b>3</b>	<b>Administrative Procedures</b>	<b>2</b>
3.1	Installation . . . . .	2
3.2	Routine Tasks . . . . .	2
3.3	Periodic Administration . . . . .	2
3.4	User Support . . . . .	2
<b>4</b>	<b>Troubleshooting</b>	<b>2</b>
4.1	Dealing with Error Messages and Failures . . . . .	2
4.2	Known Bugs and Limitations . . . . .	2
<b>A</b>	<b>Agreement Between Customer and Contractor</b>	<b>3</b>
<b>B</b>	<b>Team Review Sign-off</b>	<b>4</b>
<b>C</b>	<b>Document Contributions</b>	<b>5</b>

# 1 Introduction

Team EVAL is creating a system to more efficiently create and distribute post-semester teaching evaluations. Our client wants us to build such a system because the current one used by the University of Maine is flawed and does not meet the needs of faculty. The team must write documentation for system administrators to effectively maintain the evaluation system that is delivered by Team EVAL at the end of the semester. All members of the team plan to graduate in May 2019, and no budget exists for system maintenance.

## 1.1 Purpose of This Document

The purpose of the administrator manual is to instruct system administrators on how to properly install, maintain, and recover Team EVAL's course evaluation program. As the members of the team graduate this May, there will need to be people who know how to implement and keep the system running after the original team members leave campus. Documentation is key to onboarding new members to the project, and as such, this administrator's manual includes information about the system requirements, installation of the system, administrative tasks for the system, and troubleshooting information.

This document is intended for the development team, the product client, Dr. Harlan Onsrud, and any potential administrators of the system. The team needs it to know how to maintain the evaluation system during development. Dr. Onsrud needs the document to understand how the system is administered, and potential future administrators need it to know how to maintain the system.

## 1.2 References

Craig, J., Elliott, S., Judkins, R., & Small, S. 29 October 2018. *System Requirements Specification*.

Craig, J., Elliott, S., Judkins, R., & Small, S. 16 November 2018. *System Design Document*.

# 2 System Overview

## 2.1 Background

The system administrator will be charged with overseeing the course evaluation system, from its installation to routine maintenance. The system is composed of three major parts: the front-end user interface code written in JavaScript, the back-end API written in Python, and the MySQL database that stores survey form data.

Administrators have a variety of tasks required for a continual upkeep of the evaluations system. The most important tasks ensure that administrators will not lose access to data or to the servers. These include keeping all billing information current. If either the domain service or Amazon cannot bill the administrator, data could be lost, or access to the system could be removed. Data from both Limesurvey and the system should be backed up periodically. Backups can be stored using Amazon's Simple Storage Service (S3). While the loss of data remains unlikely, a backup should be kept and updated after each semester.

Once the system is installed, a majority of the functionality is automated, and the administrator will simply need to share the website's URL to teachers who may wish to use the software to conduct teaching evaluations.

## 2.2 Hardware and Software Requirements

The course evaluation system has low performance requirements, as the system will be hosted on a web server using Amazon Web Services, and any computer with Internet access should be able to set up and administer the course evaluation system via the console. As for memory, the system requires at least 2 GB of storage. The computer's operating system must be Windows (7 or later), Mac OS, or Linux. The evaluation system will be installed on an Amazon Web Services server instance, so administrators will need a command-line terminal such as PowerShell.

After a server instance is set up, several libraries must be installed for the system's code to run. The system requires Docker to containerize the components and LimeSurvey 3 to distribute surveys. The back

end needs MySQL to run the survey database and TaiSHiNet’s “lime-py-api” library for the back end to interface with LimeSurvey. The front end needs Node.js, the Node Package Manager with a series of node packages, and React.js for the JavaScript code to run. Most of these dependencies are installed automatically, but administrators must initially configure LimeSurvey when they install the system.

## 3 Administrative Procedures

### 3.1 Installation

The procedure to fully install the course evaluation system on an Amazon Web Services EC2 instance is listed as follows:

1. Create an Amazon Web Services account and go to the EC2 Dashboard. Click on “Instances”, then “Launch Instance”. Select the first AML, then click “Review and Launch”, then “Launch”. Select “Create a new key pair”. Name it “limesurvey”, and download the key. Click on “Launch Instances”.
2. In a terminal program, change your working directory to the folder with the key, and change the key’s permissions with the command `chmod 400 limesurvey.pem`. If this command does not work, try `$path = ".\test.txt" ,icacls.exe $path /reset, icacls.exe $path /GRANT:R "$((Get-Acl-Path .\test.txt).Owner):(R)", icacls.exe $path /inheritance:r`, replacing “test.txt” with the .pem file name. These commands give you permission to set the file as read-only.
3. In the left pane, click “Security Groups”, right-click the group with its name starting with “launch-wizard”, then click “Edit inbound rules”. Add four new rules, with their ports being 80, 5000, and 8080 respectively, and their source being “Anywhere”. Click on “Save”.
4. If using Windows 7 or 8, open PowerShell and install the “chocolatey” library found at <https://chocolatey.org/docs/installation>, then OpenSSH.
5. In the terminal, execute `ssh -i "limesurvey.pem" ec2-user@ec2-x-x-x-x.us-east-2.compute.amazonaws.com` with the x’s replaced by the fields in the instance’s IP address (see “IPv4 Public IP”). Or right-click on the instance, click “Connect”, and copy and paste the example command.
6. Enter “yes” when prompted, then run `wget -O - https://raw.githubusercontent.com/stanleysmall/capstone/master/aws.sh | bash`. After installation is finished, exit SSH with the command `exit`, then execute the SSH command again.
7. Execute `cd capstone`, then `docker-compose up -d`. After installation is finished, in your web browser, enter “teachingevaluations.org:5000”.
8. Click “Start installation”, “I accept”, and “Next”. In the database configuration, enter “10.5.0.6” for the location, “root” for the username and password, and “limesurvey” for the database name. Click “Next”, then “Create database”, then “Populate database”. Click “Next” to use the default LimeSurvey credentials.
9. Click “Administration”, and log in with a username and password of your choosing. Go to “Configuration”, then “Global settings”, then “Interfaces”. Click on “JSON-RPC”, then the toggle below. Click “Save”.
10. Execute `mysql -h 10.5.0.6 -u root -p < /home/ec2-user/capstone/sql/create_tables.sql`, entering “root” as the password.
11. Start Docker up again if it’s stopped. Test the API by entering in the web browser `http://teachingevaluations.org:8080/teameval/Eval/1.0.0/survey?name=test`. The browser should output “{}”, indicating there are no surveys in the database.

Route 53 must be configured so that `teachingevaluations.org` directs to the AWS instance. One can follow the steps at <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-to-ec2-instance.html>. This configuration must only be completed once.

## 3.2 Routine Tasks

Administrators will need to ensure that the billing information for the domain `teachingevaluations.org` is up to date. If the domain is not successfully renewed, an attacker could buy the domain and initiate phishing attacks on users. Additionally, administrators must ensure that the billing info for Amazon Web Services is up to date. Currently, the system is hosted with a "free tier" package, which allows one instance of the system to exist. For the system to be scalable during times of peak use, a paid plan is likely required. Services that are required by AWS to run the system include Elastic Cloud Compute (EC2) and Route 53 (a domain forwarding service).

Administrators should also monitor development blogs for the technologies used in this system to ensure that no vulnerabilities are discovered. Software such as LimeSurvey can be updated periodically; however, the installation script currently pulls the most recent version of LimeSurvey from the official repository.

## 3.3 Periodic Administration

To make the software maintain privacy standards and to limit liability for those administering the software, the survey data will have to be periodically deleted from the LimeSurvey database. The administrator will need to delete this data by navigating to `http://teachingevaluations.org:5000/index.php/admin/authentication/login` and following the on-screen instructions to delete the stored data. In the future, this task will be automated to reduce the need for administrator interaction as much as possible. Also, both the system database and the LimeSurvey database should be backed up periodically. Refer to the installation instructions in the installation guide for database access.

This application is self-sufficient and besides handling error messages, failures, and the periodic administration outlined above, the administrator should not need to perform any routine tasks after installing the application on their web instance.

## 3.4 User Support

Administrators can turn to available resources for help with managing the course evaluation system. The GitHub repository of the system, at `https://github.com/stanleysmall/capstone`, contains a README file that has installation instructions, as well as commands to build the code or mount the system directory on a local machine. The source code contains comments on the purpose and functionality of the code, how to format endpoint calls, and what is the expected JSON input. For personal help, one can contact the team leader, Stanley Small, through e-mail at `stanley.small@gmail.com`.

# 4 Troubleshooting

## 4.1 Dealing with Error Messages and Failures

Administrators will need to handle errors that occur during the system's operation. After accessing the web server with the SSH command, the terminal window will show error information that describes what went wrong. This is prefixed by the name of a Docker container, which lets one know which component of the system has the error. Upon an error, an administrator must first check whether he or she followed everything in the installation procedure.

If the installation procedure is followed correctly, the user should first attempt restarting the system. After accessing the web server the administrator should enter `docker-compose stop` at the command line to stop the current running process, then start the instance again by entering `docker-compose up -d`. If the problem persists, it most likely lies in the system's code. Administrators should refer the error information printed by the console to a programmer who is able to change the application's source code. Error messages state which line in a file is causing a problem, and the source code comments can help with identifying the issues. Also, Stack Overflow is a helpful website for diagnosing error messages displayed by a computer program. As a last resort, administrators should message Stanley Small with the error message.

## 4.2 Known Bugs and Limitations

Although there are no known bugs, there are a few limitations in the course evaluation system's functionality, which exist because of time constraints in system development. First, the system sends only invitation e-mails to students taking an evaluation survey. It will not send e-mails confirming survey completion or subsequent reminder e-mails to non-respondents. Second, only questions that are open-ended or answered on a 1-5 scale are permitted when creating an evaluation form. Third, there is no way to allow for a non-anonymous response that is signed by a student. Lastly, the data in both databases is not deleted automatically.

These limitations affect end users in that some students may forget to take an evaluation survey (due to lack of reminder e-mails), teacher evaluation forms are less flexible, and database data must be deleted manually. Administrators must inform potential users of the evaluation system about the limitations, and users need to take these limitations into account when managing survey forms. If necessary professors should send reminder e-mails to those taking the evaluations manually. As the framework for all of these functions is in place, an administrator would ideally employ a programmer to remove these limitations.

## A Agreement Between Customer and Contractor

This page shows that all members of Team EVAL and the client, Harlan Onsrud, have agreed on all the information in the administrator manual. By signing this document, Team EVAL and Dr. Onsrud approve the information given to system administrators about how to manage the course evaluation system correctly.

The team will follow a process in the case that the document is changed after we sign it. First, the team will write a rough draft of the changes to be made to the document. Second, all team members and Harlan Onsrud will sign the document agreeing to the changes. Finally, the team will make the changes to the final copy of the document.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
Jovon Craig	_____	_____
Sam Elliott	_____	_____
Yuanqi Guo	_____	_____
Robert Judkins	_____	_____
Stanley Small	_____	_____
Dr. Harlan Onsrud	_____	_____
Customer Comments:	_____	_____
_____		



# B Team Review Sign-off

This page shows that all members of Team EVAL have reviewed the administrator manual and agreed on its content. By signing this document, the team members agree that all information for an administrator of the course evaluation system is accurate, and there is nothing in the document that is a source of contention.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
<b>Jovon Craig</b>	_____	_____
Comments:	_____	
_____		
<b>Sam Elliott</b>	_____	_____
Comments:	_____	
_____		
<b>Yuanqi Guo</b>	_____	_____
Comments:	_____	
_____		
<b>Robert Judkins</b>	_____	_____
Comments:	_____	
_____		
<b>Stanley Small</b>	_____	_____
Comments:	_____	
_____		

## C Document Contributions

Stanley Small contributed to the routine tasks section, the periodic administration section, the purpose of this document, and the background section, and he revised the installation section. Stan contributed about 20 percent of the document.

Jovon Craig outlined the document and wrote most of the introduction, the requirements section, the installation section, the user support section, and the troubleshooting section. He also revised parts of the document. Jovon contributed about 50 percent of the document.

Yuanqi Guo made revisions and contributed to the installation section. Tom contributed about 5 percent of the document.

Sam Elliott contributed to the routine tasks, periodic administration, troubleshooting, and bugs and limitations sections. Sam contributed about 20 percent of the document.

Robert Judkins proofread the document and made changes to improve readability. Robert contributed about 5 percent of the document.

## G User Manual

# User Manual

## Course Evaluation System

Team EVAL

Jovon Craig, Sam Elliott, Yuanqi Guo, Robert Judkins, and Stanley Small

Client: Dr. Harlan Onsrud

April 18, 2019

University of Maine - Spring 2019 - COS 497  
Instructor: Professor Terry Yoo



# Course Evaluation System

## User Manual

### Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Intended Readership . . . . .	2
1.2	Applicability Statement . . . . .	2
1.3	Purpose . . . . .	2
1.4	How to Use This Document . . . . .	2
1.5	Related Documents . . . . .	3
1.6	Conventions . . . . .	3
1.7	Problem Reporting Instructions . . . . .	3
<b>2</b>	<b>Overview</b>	<b>3</b>
<b>3</b>	<b>Instructions</b>	<b>4</b>
3.1	Logging In . . . . .	4
3.2	Create Evaluation . . . . .	4
3.3	Edit Evaluation . . . . .	5
3.4	Publish Evaluation . . . . .	5
3.5	View Evaluation . . . . .	5
3.6	View Results . . . . .	5
3.7	Errors and Warnings . . . . .	5
<b>4</b>	<b>Reference Section</b>	<b>6</b>
<b>A</b>	<b>Error Messages and Recovery Procedures</b>	<b>7</b>
<b>B</b>	<b>Glossary</b>	<b>7</b>
<b>C</b>	<b>Agreement Between Customer and Contractor</b>	<b>8</b>
<b>D</b>	<b>Team Review Sign-off</b>	<b>9</b>
<b>E</b>	<b>Document Contributions</b>	<b>10</b>

# 1 Introduction

Team EVAL is creating a system to more efficiently create and distribute post-semester teaching evaluations. Our client, Dr. Harlan Onsrud, a professor at the University of Maine, wants us to build such a system. He claims that the current system used by UMaine does not meet the needs of the faculty.

The old system relies on using physical bubble sheets as evaluation surveys. Our system will allow the instructors to create a evaluation form online and distribute it via e-mail to their students. They can include whatever questions they desire and even add custom ones. Once a survey is completed, the instructor can view the results digitally, rounding off an entirely electronic process.

## 1.1 Intended Readership

This document is intended for the development team, the product client Dr. Harlan Onsrud, and any potential users and administrators of the system. The users will primarily be University of Maine faculty, such as instructors or course administrators. Other users may include instructors of courses at other universities, or anyone teaching a class outside a university setting who is interested in creating a course evaluation form.

The developers and Dr. Onsrud are most familiar with the system, so the “References” section is most relevant to them. University of Maine faculty should already be experienced with the format of the course evaluation form and will need little to no guidance to use the software. Other users should also be able to quickly pick up the software and make an evaluation form despite their unfamiliarity with its format. UMaine users may refer to the “Overview” section for a quick look at the system, and other users may refer to the “Instructions” section to learn the system’s basic functionality.

## 1.2 Applicability Statement

This user manual specifically applies to the original release of the course evaluation system, version 1.0. The document may also be applicable to future releases but will not account for the newest features.

## 1.3 Purpose

The purpose of our system is to upgrade the University of Maine’s system of processing course evaluation forms. The University has traditionally used Scantron sheets for course evaluation forms. It is tedious for students to fill in bubbles with pencil and paper. Also, school administrators must scan the forms for every student in every course, and the survey data must be collected in a form that is readable for instructors. The University of Maine has to keep up with technology by adopting a fully online solution, such as the one from Team EVAL.

This user manual is meant to provide essential information for operating our course evaluation system. It describes every function that a user can make, as well as useful procedures, warnings, and errors pertaining to the system. Overall, the software allows the process of creating, modifying, publishing, and viewing the statistics of an class evaluation form. With the user guide, one will know how to understand this process and execute it correctly and easily.

## 1.4 How to Use This Document

There are four main sections in this user manual. The first is the “Introduction” section, which introduces the readers to the system. It tells readers the target audience of the document, the purpose of Team EVAL’s system, and why the user manual was written. The second section is the “Overview” section, a general description of what the course evaluation system does. It is meant to provide a quick guide to the software. Third, there is the “Instructions” section, intended for users less familiar with the system. It details every operation possible in the system, including warnings, procedures, input, results, and possible errors for each. Finally, the “Reference” section, for those highly familiar with the application, has more technical information about the system’s operations.

Many sections of the manual relate to each other. The “Overview” section could be considered a more abstract version of the “Instructions” section. The reference section has similar information to “Instructions” but is more suitable for experienced users. A few appendix sections serve as a supplement to the main

sections. Appendix A lists all possible error messages and their suggested recovery procedures. Next, Appendix B is a glossary, defining the more obscure terms in the document.

## 1.5 Related Documents

Craig, J., Elliott, S., Judkins, R., & Small, S. 29 October 2018. *System Requirements Specification*.

Craig, J., Elliott, S., Judkins, R., & Small, S. 16 November 2018. *System Design Document*.

Craig, J., Elliott, S., Judkins, R., & Small, S. 30 November 2018. *User Interface Design Document*.

Craig, J., Elliott, S., Guo, Y., Judkins, R., & Small, S. 14 March 2019. *Code Inspection Report*.

Craig, J., Elliott, S., Guo, Y., Judkins, R., & Small, S. 4 April 2019. *Administrator Manual*.

## 1.6 Conventions

Readers of this document shall be aware of the stylistic and syntactical conventions that we use. The user is often referenced by the gender-neutral pronoun “they”. Button names and input to text fields are closed off by quotation marks. Additionally, monospaced Courier font is used for URLs, console commands, and console output.

## 1.7 Problem Reporting Instructions

Users may encounter errors in the evaluation system that cannot be easily resolved. To report any product defects or to ask questions about the software, please e-mail the project manager, Stanley Small, at stanley.small@maine.edu.

# 2 Overview

Team EVAL’s course evaluation system has multiple capabilities for instructors to efficiently manage evaluation surveys. With the software, a user may create an online survey form, use a previously created form, edit a survey form, publish a form to students, or view statistics about a survey’s responses.

The first pages the user sees upon starting the system are the landing pages. They include a summary of the system, frequently asked questions, and the log-in form. The user logs in with the e-mail address and password of a Google account. The system authenticates the log-in credentials, and the user reaches the system’s home page.

The home screen has several options for the user to take. First, one may create a new course evaluation form. By creating a new survey, the user needs to import course information such as course section, course designator, course title, course section, semester and calendar year and name of faculty unit. The second option is to edit a form that has been created but is still unpublished. Third, the user may view (but not edit) a published or inactive evaluation form. Finally, one can load a report giving statistics of the questions in course survey forms. A report applies to a particular category, such as a course section, course designator, instructor name, or university.

A major feature of the course evaluation system is editing a survey form before publishing it. If the user chose to create a new form, he or she may then select a preset form, loading the information of an existing form into the new one. While editing a form, the system allows setting the course information, including questions asking students about the course, adding custom open-ended questions, setting some questions as mandatory, listing the survey’s participants, and writing e-mail templates about completing the survey to send to students.

After editing an evaluation form to what is desired, the user may then publish the survey, look at it later, and view its statistics. Upon publishing a survey, the system imports it into a survey creation software called LimeSurvey and sends invitation e-mails to the survey’s participants. The user cannot edit a form that has been previously published. The system automatically sends reminder e-mails in case some students have not completed the survey.

An important capability of the system is to compile the responses for survey questions and output statistics of the responses. The user can view an evaluation report for a certain course section or designator, instructor last name, faculty unit, college, or university. The report gives the median, mean, standard deviation, and number of responses for 1-to-5 scale survey questions. The statistics are shown for each applicable course section, university department, and the whole university. The user may download the report as a file.

## 3 Instructions

### 3.1 Logging In

To begin using the course evaluation system, a user first needs to log in. The initial screen, known as the landing page, has a “Login” button featured on it. The only requirement to log in to the system is a valid Gmail account. Once logged in, a user will be directed to the home page, where they will have access to the functions of the evaluation system. On this page, there are four operations a user can perform at any time: “Create Evaluation”, “Edit Evaluation”, “View Evaluation”, and “View Results”.

### 3.2 Create Evaluation

When a user selects the “Create Evaluation” button, they will be directed to a new screen to begin creating a new evaluation form. This operation is used to develop a survey that will eventually be published and sent out to a list of participants. The user can either create a new form from scratch or re-use an existing form that was previously created.

If the user wants to create a new survey from scratch, they will have to first fill out the information about the course they are instructing or administering. This information includes the course designator, course title, year that the course was held, instructor’s name and e-mail address, and other details about the course and its instructor. Every field must be filled out before the user can continue. Once they are satisfied with the course details, the user can continue onto the next portion of the survey creation process.

Once the course details have been filled out, the user is now tasked with choosing what questions to include on their survey. There are several default questions that are customary in most course evaluation surveys, but the user can also create their own custom questions. For each question that appears, the user has a choice to either include it, make it mandatory for survey respondents, or exclude the question entirely.

The questions are typically split into the following categories: “Course”, “Instructor”, “Assessment”, “Teaching Assistant”, “Laboratory”, “Online”, and “Open Ended”. Some of these categories may not be relevant for a course. For example, if the course did not have a teaching assistant, then the user could disregard the “Teaching Assistant” questions entirely. Once the user has picked and created the survey questions to their liking, they can continue to the final page of creating an evaluation form.

The final page required to create a survey form is related to administration. The user is asked to list everyone enrolled in the course. The information for each student should appear in a separate row, with each row formatted as formatted as “[first name], [last name], [e-mail address]” (e.g “Mary, Smith, marysmith@gmail.com”). This section determines how the survey will be distributed once it is published. The user can also edit the e-mails these students will receive, as well as opt in or opt out of reminder e-mails to be sent to the students. Reminder e-mails are sent to the students enrolled in the course 3, 6, and 9 days after the survey is published.

Once the user is done with that page, they can finally submit the survey. Once submitted, they can choose to either save the survey to be published later or publish it right away. They will be directed back to the home page, having successfully created an evaluation form.

### 3.3 Edit Evaluation

The “Edit Evaluation” operation is used if a user has previously created a survey form but has yet to publish it. The user has the ability to select one of their existing surveys from a drop-down menu and can then choose to edit one as they please.

Once a survey is selected, the system will bring the user to a page similar to the one for creating a survey from scratch. In this situation, however, all of the fields have already been filled with what they previously entered. The user is then able to go through each page and edit the course information, the questions they have selected, and the enrollment list for the selected course. Once done, the survey can either be saved again for further editing later, or published and sent out to the participants.

### 3.4 Publish Evaluation

The "Publish Evaluation" operation is used if the user has previously created an evaluation form but has yet to publish it. The user can select one of their existing evaluations from a drop-down menu and can then choose to publish the evaluation, creating a form on LimeSurvey that will be sent to all of the survey's participants.

Once a survey is published, it can not be recalled and as such users should ensure that all information in the evaluation is accurate before publishing it. The user can view the evaluation using the "Edit Evaluation" operation listed above to ensure that they have the correct information.

### 3.5 View Evaluation

The "View Evaluation" operation is used to view an older inactive survey form that has already expired. These surveys have already been distributed and taken by the participants in a previous year or semester. Once again, the user can select the desired survey from a drop-down menu and choose which one they would like to view.

Once an evaluation form has been selected, the user is directed to a page where they can view the information from this previous course. They are unable to edit this course, as it has already been made inactive. The purpose of this function is to see the form's course information or which questions were included. At any point, the user can return to the homepage when they are done viewing.

### 3.6 View Results

The "View Results" operation displays the statistical results of a completed surveys after they have run their course. It cannot be used for older surveys from past years or semesters. Rather, the operation only applies to courses that have recently been completed and have had their surveys taken.

To choose which course evaluation results are viewed, the user needs to first select which category they would like to filter the courses: "Instructor", "Course Section", "Course Designator", "Unit", "College", and "University". Once the user selects the category type, they can then choose a more specific category. For example, if they select "Instructor", the drop-down list would consist of a list of instructors with surveys on the user's account.

Once the user has selected the category type and specific category they would like to view, they can proceed to the results screen. This screen will display the results for each question in the surveys under the designated category. It will display statistics such as the median, mean, standard deviation, and number of participants for each question. This will allow the user to compare results across courses and schools.

### 3.7 Errors and Warnings

There are two instances in which the user will be presented with an error or warning. When the user is creating or editing an evaluation form and they attempt to move on to the next page or save the form without completing all required fields, they will be prevented from doing so and presented with a red error box. This error tells the user that at least one required field in the form has not been entered. The user can recover from this by completing the required field and attempting to move on to the next page, or saving the evaluation again.

The user may encounter a warning when removing a custom question that they had previously entered. Upon pressing the "Remove" button on a custom question, the user will be prompted to confirm that they wish to remove the question. The user can either confirm the action by clicking "ok", which will cause the custom question to be removed, or select "Cancel", which does not remove the question.



## **4 Reference Section**

## **5 Error Messages and Recovery Procedures**

There is one possible error message given by the course evaluation system. If the user does not enter all required fields in a survey form, the system gives the message “ENTER MESSAGE HERE”. This error is described in the “Errors and Warnings” subsection above. The user can recover from this by completing the required field(s) and attempting to move on to the next page, or simply saving the evaluation again.

## A Glossary

- *Survey/evaluation* – The questionnaire that students enrolled in a given course will take to evaluate the course and other related entities.
- *Instructor* – The teacher of the class or course that the evaluation form is being created for. Not necessarily the same as the user.
- *Course* – The class for which an evaluation form is created.
- *Published* – A survey that has been sent to its participants so that responses may be collected.
- *Unpublished* – A survey that is not yet viewable by its participants and is still editable.
- *Active* – A survey that its users can view and for which they can make responses.
- *Inactive* – A survey that is expired and no longer viewable by its participants.
- *LimeSurvey* – An open-source software that allows the creation and publishing of surveys.
- *Back end* – The component of the system that manages survey form data and interfaces with LimeSurvey.

# B Agreement Between Customer and Contractor

This page shows that all members of Team EVAL and the client, Harlan Onsrud, have agreed on all the information in the user manual. By signing this document, Team EVAL and Dr. Onsrud approve the information given to end-users about how to operate the course evaluation system correctly.

The team will follow a process in the case that the document is changed after we sign it. First, the team will write a rough draft of the changes to be made to the document. Second, all team members and Harlan Onsrud will sign the document agreeing to the changes. Finally, the team will make the changes to the final copy of the document.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
Jovon Craig	_____	_____
Sam Elliott	_____	_____
Yuanqi Guo	_____	_____
Robert Judkins	_____	_____
Stanley Small	_____	_____
Dr. Harlan Onsrud	_____	_____
Customer Comments:	_____	_____
_____		

## C Team Review Sign-off

This page shows that all members of Team EVAL have reviewed the administrator manual and agreed on its content. By signing this document, the team members agree that all information for the user of the course evaluation system is accurate, and there is nothing in the document that is a source of contention.

<i>Name</i>	<i>Signature</i>	<i>Date</i>
<b>Jovon Craig</b>	_____	_____
Comments:	_____	
_____		
<b>Sam Elliott</b>	_____	_____
Comments:	_____	
_____		
<b>Yuanqi Guo</b>	_____	_____
Comments:	_____	
_____		
<b>Robert Judkins</b>	_____	_____
Comments:	_____	
_____		
<b>Stanley Small</b>	_____	_____
Comments:	_____	
_____		

## D Document Contributions

Stanley Small did not work on the user manual. Stan contributed 0 percent of the document.

Jovon Craig wrote part of the introduction and the whole “Overview” section. He also added the user manual template and revised much of the manual. Jovon contributed about 40 percent of the document.

Yuanqi Guo made some edits to the “Instructions” section and added a reference to another document. Tom contributed about 10 percent of the document.

Sam Elliott added to the “Instructions” section, and he documented the errors and warnings pertaining to the system. Sam contributed about 10 percent of the document.

Robert Judkins wrote the “Instructions” section, and he revised and expanded other sections. Robert contributed about 40 percent of the document.

## H Example Question Selection Form

### APPENDIX A SELECTION OF QUESTIONS AND CONTENT FOR INCLUSION ON A WEB-BASED STUDENT EVALUATION OF TEACHING FORM

#### INSTRUCTOR OR ACADEMIC UNIT SELECTION OF QUESTIONS FOR USE IN A WEB-BASED FORM:

If you want information on this form auto-filled from a previous submission that you may then edit, please **SELECT** (Note to Team: **SELECT** leads to a pulldown menu that includes all previous submission forms. When selected, all information would be copied from a previous form to the current form below except as noted.)

Course Designator (e.g. MUS)	_____
Course Number (e.g. 200)	_____
Course Section (e.g. 001)	_____
Course Title (e.g. Ballroom Dance)	_____
Is this a graduate course?	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> (bull's eye choices)
Semester and Calendar Year (e.g. Fall 2019)	_____ (don't auto populate)_
Name of Faculty Unit (e.g. School of Music)	_____
Name of College (e.g. Liberal Arts)	_____
Name of University (e.g. Univ of Maine)	_____
Last Name of Instructor (e.g. Smith)	_____ (need later in reporting to alphabetize by last name)
First Name of Instructor (e.g. Elizabeth)	_____
Instructor Email	_____
Instructor Phone	_____
Full Name of Course Evaluation Administrator	_____
Email of Course Evaluation Administrator	_____
Beginning Date of Assessments	_____ (use calendar pulldown) (don't auto populate)_
Time of Day for Any and All Mailings	_____ (use time select)_____ (don't auto populate)_
Closing Date	_____ (use calendar pulldown) (don't auto populate)_

#### NOTE TO CAPSTONE TEAM:

Variable names from some of the entries above include: {COURSEDESIGNATOR}, {COURSENUMBER}, {COURSETITLE}, {ADMINNAME}, {ADMINEMAIL}, {BEGINNINGDATE}, {CLOSINGDATE}, {TIME}  
Some other variable names likely to be used coming from the student roll upload include: {FIRSTNAME}, {LASTNAME}, {EMAIL}

# UNIVERSITY OF MAINE STUDENT EVALUATION OF TEACHING

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>THE INSTRUCTOR</b>			
How prepared was the instructor for class?	often unprepared; well prepared		
How clearly were the objectives of the course presented?	unclear; very clear		
How enthusiastic was the instructor about the subject?	very little; very much		
How clearly did the instructor present concepts, principles and theories?	unclear; very clear		
How much were you encouraged to think for yourselves?	very little; very much		
How concerned was the instructor for the quality of student learning?	unconcerned; very concerned		
Did the instructor show respect for the questions and opinions of the students?	rarely; always		
Did the instructor ensure an environment of respect for all groups of people in the classroom?	rarely; always		
Did the instructor inspire confidence in his/her knowledge?	very little; very much		
<b>Overall, how would you rate the instructor?</b>	poor; excellent		
Please list further questions to be added regarding the instructor, if any, indicating a 1-5 rating scale (i.e. 5 is best)			
(a)			
(b) Online creator should be able to add up to 15 questions			
<b>THE COURSE</b>			
Were class meetings profitable and worth attending?	rarely; always		
What is your overall rating of the primary readings?	poor; excellent		
How much did this course challenge you intellectually?	very little; very much		
How much did you learn from this course?	very little; very much		
<b>What is your overall rating of this course?</b>	poor; excellent		
Please list further questions to be added regarding the course, if any, indicating a 1-5 rating scale (i.e. 5 is best)			
(a)			
(b) Online creator should be able to add up to 15 questions			
<b>STUDENT WORK ASSESSMENT</b>			
Did the instructor let you know what was expected on the assessments (exams, assignments, projects, papers, etc.)?	unclear; very clear		
Did the assessments reflect the important aspects of the course?	rarely; always		
How fair were the grading procedures?	unfair; completely		
Overall, how would you rate the assessment process (exams, assignments, projects, papers, etc.)?	poor; excellent		
Please list further questions to be added regarding the assessment, if any, indicating a 1-5 rating scale (i.e. 5 is best)			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **laboratory questions** were copied from a previous University of Maine approved longer bubble form.

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>THE LABORATORY EXPERIENCE</b>			
Did this course have one or more regularly scheduled laboratory sessions? (If the student answer is yes, the following questions appear)			
How much did the laboratory experience contribute to your learning in this course?	very little, very much		
<b>Overall, how would you rate the laboratory experience?</b>	poor; excellent		
What was done particularly well in the laboratory experience?	----		
How could the laboratory experience be improved?	----		
Please list further lab questions to be added, if any, and a rating scale			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **teaching assistant questions** were copied from a previously used UMaine longer bubble form.

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>THE TEACHING ASSISTANT</b>			
Was there a teaching assistant supporting this course? (If answer is yes, the following questions appear)			
If there was more than one teaching assistant for the course, please name the TA you are evaluating. (open ended response)			
How much did the teaching assistant contribute to your learning in this course?	very little, very much		
How concerned was the teaching assistant for the quality of student learning?	unconcerned; very concerned		
Did the teaching assistant show respect for the questions and opinions of students?	rarely; always		
Did the teaching assistant inspire confidence in his/her knowledge?	very little; very much		
Would you want to have this teaching assistant in the future in another course?	definitely not, definitely yes		
Would you recommend this teaching assistant to assist in this course in the future?	definitely not, definitely yes		
<b>Overall, how would you rate the teaching assistant?</b>	poor; excellent		
Name something the teaching assistant did particularly well.	----		
Name something the teaching assistant could do better in the future.	----		
Please list further TA questions to be added, if any			
(a)			
(b) Online creator should be able to add up to 15 questions			



The following **online component questions** were copied from the current UMaine Augusta distance education form

Questions	1-5 Rating Scale	Include question? (If yes, indicate with )	Make response mandatory? (If yes, indicate with )
<b>ONLINE COMPONENT ASSESSMENT</b>			
Did you take this course as a distance learning student? (If answer is yes, the following questions appear)			
Please indicate the primary online modality used with the course. [only one selection allowed] <radio> Sessions were primarily <u>synchronous</u> in that I was typically required or highly encouraged to interact live online with the instructor and other students in most or all sessions. <radio> Sessions were primarily <u>asynchronous</u> in that I primarily viewed recorded video lectures or class sessions on my own schedule and had only one or fewer opportunities per week to interact live online with the instructor or other students. <radio> Sessions were approximately an equal mix of <u>synchronous</u> and <u>asynchronous</u> interactions. <radio> Other. Describe: <short text box>	----		
The online modality used with the course was well suited to my needs.	strongly disagree, strongly agree		
There was adequate opportunity for me to interact with the instructor.	strongly disagree, strongly agree		
There was adequate opportunity for me to interact with other students.	strongly disagree, strongly agree		
The online technologies used in this course worked the way they were supposed to.	strongly disagree, strongly agree		
The communication tools were easy to use (email, assignment delivery, exam delivery or proctoring, chat, blog, teleconferencing, online courseware, web, etc.).	strongly disagree, strongly agree		
Technology support was there if I needed it.	strongly disagree, strongly agree		
The online experience was well-suited to the way I like to learn.	strongly disagree, strongly agree		
Which statement best characterizes your belief after having taken this course? [only one selection allowed] <radio> I learned MUCH LESS in this online class than I probably would have in the traditional format. <radio> I learned A LITTLE LESS in this online class than I probably would have in the traditional format. <radio> I learned ABOUT THE SAME in this online class than I probably would have in the traditional format. <radio> I learned A LITTLE MORE in this online class than I probably would have in the traditional format. <radio> I learned MUCH MORE in this online class than I probably would have in the traditional format.	----		

Which statement best characterizes your belief about the grade you expect to receive in this course? [only one selection allowed] <radio> I worked MUCH LESS for my grade in this online class than I probably would have had to in the traditional format. <radio> I worked A LITTLE LESS for my grade in this online class than I probably would have had to in the traditional format. <radio> I worked ABOUT THE SAME for my grade in this online class than I probably would have had to in the traditional format. <radio> I worked A LITTLE HARDER for my grade in this online class than I probably would have had to in the traditional format. <radio> I worked MUCH HARDER for my grade in this online class than I probably would have had to in the traditional format.	----		
Please identify an e-learning aspect of the course that you found particularly valuable or beneficial	----		
Please identify an e-learning aspect of the course that could be improved.	----		
Please list further questions to be added, if any			
(a)			
(b) Online creator should be able to add up to 15 questions			

The following **open-ended questions** were copied from the current University of Maine bubble form.

Questions	1-5 Rating Scale	Include question? (If yes, indicate with ✓)	Make response mandatory? (If yes, indicate with ✓)
<b>OPEN ENDED QUESTIONS</b>			
Please identify the aspects of this course that were of most value to you.	----		
Please mention at least one additional topic or component that you would like to see included in this course.	----		
Please make any additional comments that you desire to make about the course instructor, materials or pedagogy.	----		
Please list further open-ended questions to be added, if any			
(a)			
(b) Online creator should be able to add up to 15 questions			

#### Additional Option for Mandatory Questions

For questions in the tables above that you have designated as mandatory, should the last response option for possible selection on all of these questions be “decline to respond? (If “yes”, an option 6 will be added) Yes \_\_\_ No\_\_\_

#### CLASS ROLL

Please insert in the field below the first name, last name and email of each student in the class. The information for each student should appear in a separate row and be separated by commas. (e.g. Mary, Smith, marysmith@gmail.com) You may cut and paste into the window at your option.

<window for entry – do not autopopulate in future subsequent uses of the form>

Alternatively, upload a cvs file with the content in the first row of the file being firstname, lastname, email. **UPLOAD**

#### EMAILS TO BE SENT TO STUDENTS

Initial Email invitation to Participate

Although we recommend that you do not change the following email text, you may edit the *Invitation to Participate* as appropriate for your purposes if needed. Do NOT change any item listed as a {VARIABLE} because this will cause an error in your submission that you will be forced to correct prior to successful submission. This email will be sent to each student on the begin date at the time you specified above.

<window for text – Team, Please insert from below the suggested text for DEFAULT INITIAL INVITATION TO STUDENTS>

**Do you want one or more reminder emails sent to students who have yet to respond after a few days?** Yes \_\_ No \_\_  
(If the answer is yes, the following statements and questions appear)

#### Reminder Emails

You may send up to three reminder emails to those students who have yet to complete the teaching evaluation. The text for all reminder emails will be identical.

Although we recommend that you do not change the following email text, you may edit the *Reminder Emails* to students as appropriate for your purposes if needed. Do NOT change any item listed as a {VARIABLE} because this will cause an error in your submission that you will be forced to correct prior to successful submission.

<window for text – Team, Please insert from below the suggested text for DEFAULT REMINDER TO STUDENTS>

Reminder emails are sent spaced three days apart until the closing date is reached unless you specify a longer interval.

Please send reminder emails at intervals of <insert number of 3 or above> days.

Only a maximum of three reminders will be sent regardless of the spacing. Reminder mailings are sent at the same time of day as the initial invitation mailing.

#### Confirmation Message

Although we recommend that you do not change the following email text, you may edit the *Confirmation Message* reminder to students as appropriate for your purposes if needed. Do NOT change any item listed as a {VARIABLE} because this will cause an error in your submission that you will be forced to correct prior to successful submission.

<window for text – Team, Please insert from below the suggested text for DEFAULT CONFIRMATION>

**Please double check all information provided above. Once you click SUBMIT you may not make further changes unless you receive an error notification. You will receive a summary of the student responses by email at the time and date you designated for the teaching evaluation process to end.**

IMPORTANT: You have NOT successfully completed your assessment submission until you press the SUBMIT button below AND you are taken to a web page that states your submission has been successful. If you are returned to an earlier portion of this form, complete all information items now highlighted in red and then click SUBMIT again.

#### SUBMIT

+++++

**DEFAULT INITIAL INVITATION TO STUDENTS FOR EDITING BY THE  
INSTRUCTOR/ADMINISTRATOR USER**

+++++

Subject: Invitation to Complete Evaluation for {COURSEDESIGNATOR} {COURSENUMBER} {COURSETITLE}

Dear {FIRSTNAME},

Please complete the teaching and course evaluation for {COURSEDESIGNATOR} {COURSENUMBER}  
{COURSETITLE}.

This student evaluation of teaching is completely anonymous unless you purposefully identify yourself in response to one of the questions. The software system will send you automatic reminders every few days until you complete the evaluation.

To respond, simply click the link at the end of this message.

Sincerely,  
{ADMINNAME}

-----  
Click here to complete the teaching and course evaluation:  
{SURVEYURL}

{ADMINNAME} ({ADMINEMAIL})

+++++

**DEFAULT REMINDER TO STUDENTS FOR EDITING BY THE INSTRUCTOR/ADMINISTRATOR USER**  
– TO BE SENT AT REGULAR INTERVAL UNTIL STUDENT RESPONDS, MAXIMUM OF THREE IS REACHED,  
OR CLOSING DATE IS REACHED

+++++

Subject: Reminder to Complete Evaluation for {COURSEDESIGNATOR} {COURSENUMBER} {COURSETITLE}

Dear {FIRSTNAME},

Recently we invited you to complete a teaching evaluation for “{COURSEDESIGNATOR} {COURSENUMBER} {COURSETITLE}”. We note that you have not yet completed the evaluation, and wish to remind you that it is still available should you wish to take part.

To participate, please click on the link below.

The deadline for completing the evaluation is {CLOSINGDATE} at {TIME}.

Your participation is extremely important to the improvement of teaching and courses at the {UNIVERSITYNAME}

Sincerely,  
{ADMINNAME}

-----  
Click here to complete the teaching and course evaluation:  
{SURVEYURL}

**NOTE TO TEAM: SHOULD THE FOLLOWING GREEN OPTION BE INCLUDED AND IMPLEMENTED?**

If you do not want to participate in the student evaluation process and don't want to receive any more email reminders, please click the following link:

{OPTOUTURL}

Note: Such an option is required in many instances by “Do Not Spam” federal legislation but would not be required typically in a university mailing to your own students. Implement as good practice regardless?

{ADMINNAME} ({ADMINEMAIL})

+++++

**DEFAULT CONFIRMATION**

+++++

Subject: Completion of Course Evaluation for {COURSENUMBER} {COURSETITLE}

Dear {FIRSTNAME},

This email is to confirm that you have completed a student evaluation of {COURSEDESIGNATOR}  
{COURSENUMBER} {COURSETITLE} at the {UNIVERSITYNAME}. Your response has been saved. Thank you for participating.

If you have any further questions about this email, please contact {ADMINNAME} at {ADMINEMAIL}.

Sincerely,

{ADMINNAME}

# I Example Results Display

## REPORT FOR INSTRUCTOR: ROY TURNER

(includes all courses taught by this instructor this semester under this administrator account)

Fall 2018
University of Maine (UMaine)
College of Liberal Arts and Sciences (CLAS)

### THE INSTRUCTOR

How prepared was the instructor for class? (1- often unprepared; 5- well prepared)				
	Median	Mean	Std. Dev	n
COS 120 (Sec 003) Instr: Turner				
COS 490 (Sec 001) Instr: Turner				
INT 601 (Section 002) Instr: Turner				
COMPARATIVE UNDERGRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All COS undergraduate courses (submitted under this admin account for this semester)				
All SCIS undergraduate courses (Faculty Unit - this admin account for this semester)				
All CLAS undergraduate courses (College - this admin account for this semester)				
All UMaine undergraduate courses (University - this admin account for this semester)				
COMPARATIVE GRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All INT graduate courses (submitted under this admin account for this semester)				
All SCIS graduate courses (Faculty Unit - this admin account for this semester)				
All CLAS graduate courses (College - this admin account for this semester)				
All UMaine graduate courses (University - this admin account for this semester)				

How clearly were the objectives of the course presented? (1- unclear; 5- very clear)				
	Median	Mean	Std. Dev	n
COS 120 (Sec 003) Instr: Turner				
COS 490 (Sec 001) Instr: Turner				
INT 601 (Section 002) Instr: Turner				
COMPARATIVE UNDERGRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All COS undergraduate courses (submitted under this admin account for this semester)				
All SCIS undergraduate courses (Faculty Unit - this admin account for this semester)				
All CLAS undergraduate courses (College - this admin account for this semester)				
All UMaine undergraduate courses (University - this admin account for this semester)				
	Median	Mean	Std. Dev	n
COMPARATIVE GRADUATE COURSE RESULTS				
All INT graduate courses (submitted under this admin account for this semester)				
All SCIS graduate courses (Faculty Unit - this admin account for this semester)				
All CLAS graduate courses (College - this admin account for this semester)				
All UMaine graduate courses (University - this admin account for this semester)				

Etc.

Overall, how would you rate the instructor? (1- poor; 5- excellent)				
	Median	Mean	Std. Dev	n
COS 120 (Sec 003) Instr: Turner				
COS 490 (Sec 001) Instr: Turner				
INT 601 (Section 002) Instr: Turner				
COMPARATIVE UNDERGRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All COS undergraduate courses (submitted under this admin account for this semester)				
All SCIS undergraduate courses (Faculty Unit - this admin account for this semester)				
All CLAS undergraduate courses (College - this admin account for this semester)				
All UMaine undergraduate courses (University - this admin account for this semester)				
	Median	Mean	Std. Dev	n

COMPARATIVE GRADUATE COURSE RESULTS				
All INT graduate courses (submitted under this admin account for this semester)				
All SCIS graduate courses (Faculty Unit - this admin account for this semester)				
All CLAS graduate courses (College - this admin account for this semester)				
All UMaine graduate courses (University - this admin account for this semester)				

Note that the following is a **Further Question** and thus the question is **not** compared with other grouped evaluations

<b>Is this instructor well groomed?</b> (1- never; 5- always)				
	Median	Mean	Std. Dev	n
COS 120 (Sec 003) Instr: Turner				
COS 490 (Sec 001) Instr: Turner				
INT 601 (Section 002) Instr: Turner				

Etc.

#### THE COURSE

<b>Were class meetings profitable and worth attending?</b> (1- rarely; 5- always)				
	Median	Mean	Std. Dev	n
COS 120 (Sec 003) Instr: Turner				
COS 490 (Sec 001) Instr: Turner				
INT 601 (Section 002) Instr: Turner				
COMPARATIVE UNDERGRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All COS undergraduate courses (submitted under this admin account for this semester)				
All SCIS undergraduate courses (Faculty Unit - this admin account for this semester)				
All CLAS undergraduate courses (College - this admin account for this semester)				
All UMaine undergraduate courses (University - this admin account for this semester)				
COMPARATIVE GRADUATE COURSE RESULTS	Median	Mean	Std. Dev	n
All INT graduate courses (submitted under this admin account for this semester)				



All SCIS <u>graduate</u> courses (Faculty Unit - this admin account for this semester)				
All CLAS <u>graduate</u> courses (College - this admin account for this semester)				
All UMaine <u>graduate</u> courses (University - this admin account for this semester)				

Etc.

## OPEN ENDED QUESTIONS

Please identify the aspects of this course that were of most value to you.	
	Volunteered Self Identification by a Student and Verified
<b>COS 120 (Sec 003) Student Comments</b>	
The jokes.	Anonymous
Instructor really knew the material.	Jovon Adams
The exercises were great.	Anonymous
<b>COS 490 (Sec 001) Student Comments</b>	
The readings were marvelous	Anonymous
It provided a great hour for napping	Anonymous
The insights of my student peers.	Sam Sneed
<b>INT 601 (Section 002) Student Comments</b>	
This course sucked.	Anonymous
The snacks were good.	Betty Snow
The case studies were the most important part of the course.	Anonymous
The homework exercises were great.	Anonymous

Etc.