

# 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



# 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>3</b>
3.1	Installation . . . . .	3
3.2	Routine Tasks . . . . .	4
3.3	Periodic Administration . . . . .	4
3.4	User Support . . . . .	4
<b>4</b>	<b>Troubleshooting</b>	<b>4</b>
4.1	Dealing with Error Messages and Failures . . . . .	4
4.2	Known Bugs and Limitations . . . . .	5
<b>A</b>	<b>Agreement Between Customer and Contractor</b>	<b>6</b>
<b>B</b>	<b>Team Review Sign-off</b>	<b>7</b>
<b>C</b>	<b>Document Contributions</b>	<b>8</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, in his opinion, the current system used by UMaine does not optimally meet the needs of the 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 AMI, 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, there are no help buttons in the user interface. 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 they must consult the user manual for help, 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. As the framework for the missing 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.