PROJECT TANGELO

TEAM 3

This Page Intentionally Left Blank

TABLE OF CONTENTS

This Page Intentionally Left Blank

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Libero animi molestiae ea sunt consequuntur nemo possimus dolore temporibus, aperiam sed ut eum alias mollitia voluptatum. Molestias adipisci praesentium sapiente assumenda.

1.2 STATEMENT OF PROBLEM AREA

The purpose of this project was to develop an application that would allow individuals, such as teacher or professors, to manage isolated container instances, that would in turn be accessed via a browser by users, such as students.

1.3 BRIEF PROJECT DESCRIPTION

Our team would like to create an application that allows user to login to a Linux terminal session through the browser. The user's session will be sandboxed and he or she will be able to have full control over their container instance. An important feature of our application is that it will be easy to install by an administrator so that he or she may use it on their own servers.

The administrator may also provide documents for things like a lesson plan or assignments that other users may refer to while using the system. Another important feature would be allowing the administrator to control each container a user has. If a user locks up their container, the administrator will be able to halt and reset it. An application such as ours can be useful for small vocational schools or secondary level institutions where appropriating a large IT staff is not feasible.

1.4 PURPOSE/OBJECTIVES

Our product would allows users to have hands on experience with the use Linux through a web browser, in situations where they would otherwise not have access to. In addition, by giving administrators the ability to create, destroy, halt, or restart instances will allow users to recover from what could have been a fatal mistake.

Finally, by allowing instances to be kept separate and/or on their own server(s), the actions of one user will not negatively impact other users.

1.5 TEAM ACHIEVEMENTS

SPRINT 1

FEBURARY 26TH, 2015

User Stories	Tasks	Responsible Party
Administrator will be able to set a user's password Administrator will be able to set password expiration Users will be able to request to reset their passwords	Write user authentication Write user login page Write dashboard functionality Design views and forms for modifying user passwords and dashboard	Shane Satterfield Tyler Goodman
Administrator will be able to set up user accounts from a CSV file Administrator will be able to set up to create and remove individual users from an administrator panel	Design views and forms for uploading CSV files and managing users Write functionality for parsing a CSV file and creating those users it Write user model with required fields	Eduardo Arevalos
Users will be able to manage their accounts and profile (for example name, email address and password)	Design views and forms for managing a user profile Write functionality for committing those changes to the database	Tiffany Artis
User will be able have full access control over their container instance (including installing things inside it)	Setup virtual machines for development Create LXC templates that will be a user's default system	David Nuon Forest Turner

User will be able to log into the system (from the browser) and access a terminal session that is logged into their container instance	Write automatic login into container from browser Design view that will hold the terminal and user menu Embed the terminal in user view	Shane Satterfield Johnny Patterson

SPRINT 2

MARCH 19TH, 2015

User Stories	Tasks	Responsible Party
Administrator will be able to submit documents (PDF, HTML, DOCX, and ODF) for other users to view	Write file handling and functionality for associating with lessons Design and write views for managing documents	Shane Satterfield Tyler Goodman Tiffany Artis Eduardo Arevalos
User will be able to have a text scratchpad available to take notes.	Write widget and button toggle that hides and shows scratchpad	Eduardo Arevalos
Administrator will be able to set quotas for usage for the containers (CPU shares, memory usage) Administrator will be able to shutdown and restart containers if it is frozen Administrator will be able to view the status of servers and container instances from a control panel	Write utility to manage container quotas so it can be used by node Write utility to shutdown and restart instances Write utility to poll other servers Design and write views so they can interact with the utilities	David Nuon Forest Turner Shane Satterfield Tyler Goodman
SSL	Setup and configure SSL	Johnny Patterson
User will be able to log into a terminal session from the browser	Implement user terminal using WebSockets instead of AJAX	Johnny Patterson

SPRINT 3

APRIL 9TH, 2015

User Stories	Tasks	Responsible Party
User will be able to submit work to the administrator	Implement functionality that allows users to submit work from inside their container Implement functionality to show the result in the user's dashboard	Eduardo Arevalos Tyler Goodman
Administrator will be able to view work a user submitted and provide feedback on user submissions	Implement functionality to view user submissions and comment on them	Shane Satterfield Tiffany Artis
Administrator will be able to add/remove container instances	Implement saltstack module to send add/remove instance commands to other servers	David Nuon Forest Turner
Administrator will be able to deploy the software from an automatic install Administrator will be able to update the software automatically	Write installer for the node.js frontend for the system Write saltstack configuration files so that other servers will be able to be configured with the proper packages Test on Vagrant boxes and DigitalOcean	Johnny Patterson David Nuon

FINAL SPRINT

MAY 14, 2015

User Stories	Tasks	Responsible Party
User will find minimal errors	Fix any bugs / regressions that haven't been addressed in previous sprints	Entire Team
Bonus features	Choose and implement one of the stretch goals if time allows	Entire Team

CHAPTER 2

SYSTEM FUNCTIONAL SPECIFICATION

2.1 FUNCTIONS PERFORMED

2.2 USER INTERFACE DESIGN

2.3 USER INPUT PREVIEW

2.4 USER OUTPUT PREVIEW

2.5 SYSTEM DATABASE/FILE STRUCTURE PREVIEW

2.6 EXTERNAL AND INTERNAL LIMITATIONS AND RESTRICTIONS

USER INTERFACE SPECIFICATION

This Page Intentionally Left Blank

CHAPTER 3

SYSTEM PERFORMANCE REQUIREMENTS

3.1 RELIABILITY

The software should be deployed on a platform that is able to spawn and image server instances, like Digital Ocean or Amazon EC2.

3.2 SECURITY

HTTPS is required for the browser. All transations with Salt are done with AES encryption.

3.3 MODIFIABILITY

The source is freely available at https://github.com/ProjectTangelo

3.4 PORTABILITY

Linux is required for running the system.

3.5 OTHER

3.5.1 SOFTWARE REQUIREMENTS

Ubuntu 14.10 is required. The installer will take care of all dependencies.

3.5.2 BROWSERS SUPPORTED

Chrome, Firefox and Safari as supported.

CHAPTER 4

SYSTEM DESIGN OVERVIEW

4.1 SYSTEM DATA FLOW DIAGRAMS

4.2 SYSTEM STRUCTURE CHARTS

4.3 SYSTEM DATA DICTIONARY

4.4 SYSTEM INTERNAL DATA STRUCTURE PREVIEW

4.5 DESCRIPTION OF SYSTEM OPERATION

4.6 EQUIPMENT CONFIGURATION

4.7 IMPLEMENTATION LANGUAGES

REQUIRED SUPPORT SOFTWARE

This Page Intentionally Left Blank

CHAPTER 5

SYSTEM DATA STRUCTURE SPECIFICATIONS

5.1 OTHER USER INPUT SPECIFICATION

5.2 OTHER USER OUTPUT SPECIFICATION

5.3 SYSTEM DATABASE/FILE STRUCTURE SPECIFICATION

SYSTEM INTERNAL DATA STRUCTURE SPECIFICATION

This Page Intentionally Left Blank

CHAPTER 6

MODULE DESIGN SPECIFICATION

6.1 MODULE FUNCTIONAL SPECIFICATION

MODULE OPERATIONAL SPECIFICATION

This Page Intentionally Left Blank

CHAPTER 7

SYSTEM VERIFICATIONS

7.1 ITEMS/FUNCTION TO BE TESTED

7.1.1 INSTALLING AND MANAGING SOFTWARE

7.1.2 PASSWORD MANAGEMENT

7.1.3 ACCOUNT MANAGEMENT

7.1.4 INSTANCE MANAGEMENT

7.1.5 LESSON PLANS AND ASSIGNMENTS

7.1.6 TERMINAL SESSIONS

7.2 DESCRIPTION OF TEST CASES

7.3 JUSTIFICATION OF TEST CASES

7.4 TEST RUN PROCEDURE AND RESULTS

7.4.1 VISUAL DIFF/PHANTOMJS TESTING

7.4.2 PROTRACTOR TESTING

7.4.3 OTHER KIND OF TESTING

7.4.4 OTHER KIND OF TESTING

7.4.5 OTHER KIND OF TESTING

7.5 DISCUSSION OF TEST RESULTS

Test Case Number	Testcase-01
Test Item	Admin will be able to successfully login
Pre-Conditions	Website must be active
Post-Conditions	Successfully Logged in as Admin
Input Specifications	Username and password
Expected Output Specifications	Redirected to admin dashboard
Pass/Fail Criteria	After login client is taken to admin dashboard
Assumptions and Constraints	Website is active and working properly
Dependencies	asdf

Test Case Number	Testcase-02

Test Item	User will be able to successfully login
Pre-Conditions	Website must be active
Post-Conditions	Successfully Logged in as user
Input Specifications	Username and password
Expected Output Specifications	Redirected to user container/dashboard
Pass/Fail Criteria	After login client is taken to user container/dashboard
Assumptions and Constraints	Website is active and working properly
Dependencies	asdf

Test Case Number	Testcase-03
Test Item	Admin can add a user

Pre-Conditions	Must be logged in as an Admin
Post-Conditions	Added user appears in user lists
Input Specifications	desired username, first name, last name, e-mail address, user-type(user or admin), password, and password confirmation
Expected Output Specifications	There is a new user
Pass/Fail Criteria	User can be seen in list of users
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-04
Test Item	Admin can add multiple users a user

Pre-Conditions	Must be logged in as an Admin
Post-Conditions	Added user appears in user lists
Input Specifications	CSV file that contains valid information for the the users to be added
Expected Output Specifications	There are new users
Pass/Fail Criteria	Users can be seen in list of users
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-05
Test Item	Admin can edit a user’s information
Pre-Conditions	Must be logged in as an Admin

Post-Conditions	Change in user information appears in user lists
Input Specifications	desired change username, first name, last name, e-mail address, user-type(user or admin), or password and password confirmation
Expected Output Specifications	User's information has been edited
Pass/Fail Criteria	Change in user information can be seen in list of users
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-06
Test Item	Admin can delete a user
Pre-Conditions	Must be logged in as an Admin

Post-Conditions	Deleted user no longer appears in user lists
Input Specifications	User selected to be deleted
Expected Output Specifications	User has been deleted
Pass/Fail Criteria	Deleted user no longer appears in user lists
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-07
Test Item	Lesson can be added by admin
Pre-Conditions	Must be logged in as admin
Post-Conditions	Lesson appears on lesson list
Input Specifications	Lesson/file to be added
Expected Output Specifications	Lesson is added

Pass/Fail Criteria	Lesson appears on lesson list
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-07
Test Item	Lesson can be added by admin
Pre-Conditions	Must be logged in as admin
Post-Conditions	Lesson appears on lesson list
Input Specifications	Lesson/file to be added
Expected Output Specifications	Lesson is added
Pass/Fail Criteria	Lesson appears on lesson list
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-08
Test Item	Lesson can be deleted by admin
Pre-Conditions	Must be logged in as admin
Post-Conditions	Lesson no longer appears on lesson list
Input Specifications	Lesson to be deleted
Expected Output Specifications	Lesson is removed
Pass/Fail Criteria	Lesson no longer appears on lesson list
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-09
Test Item	Lesson can be seen/opened by user
Pre-Conditions	Lesson has been added

	Logged in as a user
Post-Conditions	Can view lesson text
Input Specifications	Select lesson to view/open
Expected Output Specifications	Lesson/file text appears onscreen
Pass/Fail Criteria	Lesson/file text for selected lesson/file appears onscreen
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-10
Test Item	User can submit work to Admin
Pre-Conditions	Must be logged in as user There must be a lesson to submit work for?

Post-Conditions	User work has been successfully submitted
Input Specifications	Work to be submitted
Expected Output Specifications	User work has been submitted
Pass/Fail Criteria	User work has been successfully submitted
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-11
Test Item	Admin can see user submission
Pre-Conditions	A user submission exists Must be logged in as Admin
Post-Conditions	Admin can view user submission
Input Specifications	Submission to view

Expected Output Specifications	Submission is displayed on screen
Pass/Fail Criteria	Selected User submission can be viewed and is displayed onscreen
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-12
Test Item	Admin can give feedback to user
Pre-Conditions	Must be logged in as admin There must exist a submission for which to give feedback
Post-Conditions	Feedback has been posted
Input Specifications	Text of feedback to be given
Expected Output Specifications	Feedback had been submitted

Pass/Fail Criteria	Feedback has been successfully submitted for desired user submissions
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-13
Test Item	User can see feedback given by Admin Feedback must exist
Pre-Conditions	Must be logged in as a User
Post-Conditions	Feedback can be viewed by user
Input Specifications	Desired feedback to view
Expected Output Specifications	Feedback is displayed for User
Pass/Fail Criteria	Desired feedback is viewable and displayed to User

Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-14
Test Item	Administrator will be able to deploy the software from an automatic install
Pre-Conditions	asdf
Post-Conditions	asdf
Input Specifications	asdf
Expected Output Specifications	asdf
Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf

Dependencies	asdf

Test Case Number	Testcase-15
Test Item	Administrator will be able to view the status of servers and container instances from a control panel
Pre-Conditions	asdf
Post-Conditions	asdf
Input Specifications	asdf
Expected Output Specifications	asdf
Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-16
Test Item	Administrator will be able to add and remove container instances
Pre-Conditions	asdf
Post-Conditions	asdf
Input Specifications	asdf
Expected Output Specifications	asdf
Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-17

Test Item	Administrator will be able to freeze and restart container instances from a control panel
Pre-Conditions	asdf
Post-Conditions	asdf
Input Specifications	asdf
Expected Output Specifications	asdf
Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-18
Test Item	Administrator will be able to shutdown and restart containers if it is frozen

Pre-Conditions	asdf
Post-Conditions	asdf
Input Specifications	asdf
Expected Output Specifications	asdf
Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-19
Test Item	User will be able to have a text scratchpad available to take notes
Pre-Conditions	Must be logged in as a User
Post-Conditions	Scratchpad must be available for use

Input Specifications	Click on button to bring up the scratchpad
Expected Output Specifications	Scratchpad is now displayed on screen
Pass/Fail Criteria	Scratchpad has successfully displayed onscreen and can be used by User
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-20
Test Item	Administrator will be able to update the software automatically
Pre-Conditions	asdf
Post-Conditions	asdf
Input Specifications	asdf

Expected Output Specifications	asdf
Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-21
Test Item	Administrator will be able to set a user’s password
Pre-Conditions	asdf
Post-Conditions	asdf
Input Specifications	asdf
Expected Output Specifications	asdf

Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-22
Test Item	Administrator will be able to set how long until a user needs to change their password
Pre-Conditions	asdf
Post-Conditions	asdf
Input Specifications	asdf
Expected Output Specifications	asdf
Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf

Dependencies	asdf

Test Case Number	Testcase-23
Test Item	Users will be able to request to reset their passwords if they have forgotten them
Pre-Conditions	Must be attempting to log into a User account User cannot log in
Post-Conditions	User will be able to successfully log in
Input Specifications	User inputs username and new password
Expected Output Specifications	User is logged into their account on the server
Pass/Fail Criteria	User is able to get past login screen and view their homepage
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-24
Test Item	Users will be able to manage their accounts and profile (for example name, email address and password)
Pre-Conditions	Must be logged in as User User able to view their homepage and profile
Post-Conditions	Changes made to profile will be saved and displayed
Input Specifications	User inputs different information than currently provided
Expected Output Specifications	New information posted to user profile
Pass/Fail Criteria	Users’ new information is displayed upon viewing the profile
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-25
Test Item	User will be able have full access control over their container instance (including installing things inside it)
Pre-Conditions	Must be logged in as User
Post-Conditions	User able to install required components, as well as make changes to and save lessons
Input Specifications	asdf
Expected Output Specifications	asdf
Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-26
Test Item	Administrator will be able to set quotas for usage for the containers (CPU shares, memory usage)
Pre-Conditions	asdf
Post-Conditions	asdf
Input Specifications	asdf
Expected Output Specifications	asdf
Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf
Dependencies	asdf

Test Case Number	Testcase-27

Test Item	User will be able to adjust the sizes of the panels in their terminal session (terminal, lesson plan, scratchpad)
Pre-Conditions	asdf
Post-Conditions	asdf
Input Specifications	asdf
Expected Output Specifications	asdf
Pass/Fail Criteria	asdf
Assumptions and Constraints	asdf
Dependencies	asdf

CHAPTER 8

CONCLUSIONS

8.1 SUMMARY

8.2 PROBLEMS ENCOUNTERED AND SOLVED

8.3 SUGGESTION FOR BETTER APPROACHES TO PROBLEM/PROJECT

SUGGESTIONS FOR FUTURE EXTENSIONS TO PROJECT

This Page Intentionally Left Blank

CHAPTER 9

APPENDIX

This Page Intentionally Left Blank

CHAPTER 10

PROGRAM LISTINGS

This Page Intentionally Left Blank

CHAPTER 11

API

11.0.1 GENERAL FORMAT

Most of the resources defined in the API schemas follow the same general outline:

11.0.1.1 GET /:RESOURCE

Returns all resources that the given user is allowed to see

11.0.1.2 GET /:RESOURCE/:ID

Returns the resource that belongs to that identifier

11.0.1.3 GET /:RESOURCE?CONDITIONS={}&FIELDS=""&OPTIONS={}

Returns all resources that pass the given criteria - Query parameters: - conditions - JSON literal - defines the search criteria for the returned resources - fields - space-delimited string of field names - defines the fields in the resource that are returned - options - JSON literal - mongodb query options

11.0.1.4 POST /:RESOURCE

Creates a resource. Fields are defined in the POST urlencoded body.

11.0.1.5 PUT /:RESOURCE/:ID

Updates a value of a resource. Updated values are defined in the POST urlencoded body.

11.0.1.6 DEL /:RESOURCE/:ID

Deletes a resource. Often requires administrator authorization.

11.0.2 AVAILABLE RESOURCES

- /user
- /submission
- /feedback
- /file
- /upload

11.0.2.1 /UPLOAD

The **/upload** resource is unique from the others. Instead of providing a model and CRUD service on that model, **/upload** provides a means to retrieve previously uploaded files on the **/file** resource. The **/file** resource itself stores and returns metadata for a given file.