Test Plan

Project Name: Tresearch
Application Type: Web Application

Trial By Fire
Jessie Lazo
Matthew Chen (Team Lead)
Pammy Poor
Viet Nguyen

Instructor: Vatanak Vong

Submission Date
10/27/2021

Updated Date
12/13/2021

California State University, Long Beach
College of Engineering

1. **Introduction**
   - **1.1 Purpose**

     This test plan documents the approach and methodologies used to ensure that each application feature meets the full requirements and satisfies the use case scenarios listed in the business requirement document.
   - **1.2 Project Overview**

     Tresearch is an interactive mind mapping tool for documenting users' learning journey. Users can create their own knowledge tree(s) to document what they are currently learning and have learned throughout their life. Knowledge trees are made up of nodes that consist of a title, description/summary, and optional tag. Branches that come off of nodes point to nodes that utilize or require knowledge from the previous node. Users can view any other user's public knowledge trees to see what they are learning or have learned and how. If a user finds another user's tree or a portion of their tree to be particularly useful or helpful, they can rate a particular node or section of the tree and they can also copy that section over to their own tree. Users can add additional information to their public profiles such as what they are currently learning, and where they are working/what they currently do or have done. Users can utilize a search function in order to find people whose trees contain a certain topic or keyword/phrase, and can also utilize a filter to narrow searches by users who are learning said topic, are using said topic (i.e. in their work or otherwise), by rating, and by tags.
   - **1.3 References**

     Documents that will be referenced include:
     - Business Requirements Document
     - Technical Specifications
2. **Scope**
   - **2.1 Features to be tested**

     The features to be tested include:
     - Creating a root node
     - Creating new node(s)
     - Copying a node(s)
     - Pasting a node(s)
     - Setting a node(s) private/public state
     - Changing the contents of a node
     - Changing the parent of a node
     - Deleting node(s)
     - Searching for a topic
       - Applying filters before searching
       - Applying filters after searching
     - Rating node(s)

2.2 Core Components to be tested

The core components to be tested include:

- Securit
    - Authentication
    - Authorization
    - Logout
- User Administration
    - Account Creation (Registration)
    - Account Deletion
    - User Management
        - Create Account
        - Delete Account
        - Update Account
        - Enable Account
        - Disable Account
    - Usage Analysis Dashboard
    - Logging
    - Archiving
    - User Privacy
    - Error Handling
    - UI/UX

3. Testing Environment

Testing will be done using the latest version of the supported browser, Google Chrome v.94.0.4606.71+, as noted in the Technical Specifications documented. The hardware and operating system will go as follows:

- Processor - Intel(R) Core™ i7-8750H CPU at 2.20 GHz
- RAM - 16 GB DDR4-2667 MHz Dual Channel
- Storage - Samsung 970 EVO Plus 1 TB NVMe SSD (Read 3,500 MB/s, Write 3,300 MB/s)
- Windows 10 Home 64 bit

Tools to be used include xUnit for automatic unit testing.

4. Testing Criteria

4.1 Exit Criteria

In order to meet the exit criteria, each test case must meet a 100% run rate as well as a 95% pass rate.

4.2 Suspension Criteria

If testing has a failure rate of more than 50%, testing will be suspended until bugs identified are fixed by developers. Testing will be restarted once the bugs identified are fixed.

5. **Testing Type**

5.1 Unit Testing

During the development phase of each application feature, the developer will test that each feature performs as expected. Unit testing ensures that bugs are fixed early in development. Unit tests will be done manually automatically using NUnit.

5.2 Integration Testing

After code is pieced together, developers will perform testing done on each feature. This is to make sure that each feature cohesively works together. These tests are the same done in the unit tests of each feature but done so when all code is working together.

5.3 System Testing

Testing will be done on features not worked by testers that did not have a part in writing the specific feature. These tests will be done both manually and automated. Automated testing will be done using NUnit.

6. **Test Logistics**

Task	Members	Estimated Effort (hours)
Create Test Scripts	Test designer (cannot be individual's code)	30 hours
Perform Tests	Tester	5 hours
Test Report	Tester	10 hours

7. **Roles and Responsibilities**

7.1 Developer

- Develop individual features that align with specific use cases
- Create Scripts used for unit testing
- Conduct Unit Testing
- Produce Report for Unit Testing

7.2 Test Designer

- Must not be developer for specific feature
- Create scripts for system testing

7.3 Tester

- Conduct System Test scripts
- Produce Report for System Tests

8. **Test Schedule**

Feature/Component	Estimated Effort Points	Expected Starting Date	Expected Completion Date
Creating Nodes	5 Hours	2/18/2022	3/2/2022
Copying Nodes	5 Hours	3/18/2022	3/30/2022
Pasting Nodes	5 Hours	4/1/2022	4/13/2022
Setting Privacy Modes	3 Hours	3/4/2022	3/16/2022
Editing a Node	2 Hours	4/1/2022	4/13/2022
Editing Node Tags	2 Hours	4/1/2022	4/13/2022
Changing a Parent Node	6 Hours	4/15/2022	4/27/2022
Deleting Nodes	5 Hours	3/4/2022	3/16/2022
Searching	6 Hours	3/4/2022	3/16/2022
Node Rating	2 Hours	4/1/2022	4/13/2022
Authentication	4 Hours	11/20/2021	12/15/2021
Authorization	3 Hours	11/20/2021	12/15/2021
Logout	3 Hours	2/18/2022	3/2/2022
Account Creation (Registration)	3 Hours	2/18/2022	3/2/2022
Account Deletion	2 Hours	2/4/2022	2/16/2022
User Management	5 Hours	11/20/2021	12/15/2021
Usage Analysis Dashboard	6 Hours	4/1/2022	4/13/2022
Logging	5 Hours	11/20/2021	12/15/2021
Archiving	2 Hours	11/20/2021	12/15/2021
User Privacy	2 Hours	2/18/2022	3/2/2022
Error Handling	2 Hours	2/4/2022	2/16/2022
UI/UX	2 Hours	3/4/2022	3/16/2022
Total Time To Execute Test Plan	41 Hours	2/18/2022	4/13/2022

9. Test Cases

9.1 Creating a Root Node (Estimated 2.5 hours)

Purpose

Verify that a user can create a root node. A root node is a brand new tree and has no children.

Preconditions

- User is an active and authenticated user
- User has a stable internet connection
- User has navigated to the user's tree portal
- Number of nodes created by the user has not reached the hard limit

Requirements

- User enters valid name
    - Valid node name consists of
        - Minimum of 1 character and are limited to 50 characters
        - Unique in the user's account. The name must not already exist within any of the user's nodes
    - Valid node descriptions consist of:
        - Minimum of 1 character and are limited to 300 characters
- User selects valid tags
    - Valid tags are found from a bank of tags provided
    - A created node does not require a tag
- User selects valid privacy mode
    - Privacy toggles between private and public
        - By default a root node will be set public
        - Any of the root node's children will inherit it's privacy feature
        - A child cannot be public if it's parent is privacy
- A maximum of 150 nodes can be created or pasted within 24 hours with the same account before a hard limit is reached and node creation is disabled
- A maximum of 10 nodes can be created or pasted before a user reaches a soft limit.

Specific Input Example

Name: Underwater Basket Weaving
Description: A course considered by many to be the most difficult capstone course offered at CSULB
Privacy: Public
Tags: Course

Test Steps

1. Click on *Create A Tree* button
2. If user has reached soft limit, user will
    a. Solve captcha
    b. Click *Ok* button
3. User enters node name into *Name* input field

4. User enters node description into *Description* input field
5. User toggles between privacy modes desired
6. If the user desires to add tags
    a. Click *add tags*
    b. User searches for relevant tags in search bar
    c. Click on tag to add
    d. Click *Close*
7. Click the *Save* button

Success Cases

- User creates and saves a root node with valid inputs. The user is automatically navigated to a page containing an overview of the created tree.
- User creates and saves a root node with valid inputs. User meets the soft limit and successfully solves captcha. The user is automatically navigated to a page containing an overview of the created tree.

Fail Cases

- User creates and saves a root node with valid inputs. The automatic navigation does not take place.
- User creates and saves a root node with valid inputs. User is navigated to a page other than the new tree's overview.
- User creates and saves a root node with valid inputs. Database connection is broken. System message displays "This feature is currently undergoing maintenance. Please try again at a later time."
- User creates and saves a root node with empty inputs. System message displays "Input field cannot be left empty."
- User attempts to create a root node after the user's hard limit has been reached. System message displays "You have reached your limit for nodes today. The node was not created. Please delete some nodes or try again tomorrow."
- User creates and saves a root node with a name that already exists in the user's portal. System message displays "There is already a node with that title."

Non-functional Requirements

- New page containing node's input fields is loaded within five seconds
- User tree database is updated within 5 seconds
- User's hard limit count is incremented by one
- User's soft limit count is incremented by one
- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.2 Creating a Node (Estimated 2.5 Hours)

Purpose:

Verify that a user can create a node in an existing tree

Preconditions:

- User is active authenticated user
- User has a stable internet connection
- User has navigated to the user's tree portal with an existing tree.
- User knows which node will be the parent node
- Number of nodes created by the user has not reached the hard limit

Requirements:

- User enters valid name and descriptions in relative input fields
  - Valid node names consist of:
    - minimum of 1 character and are limited to 50 characters.
    - Unique. The name must not already exist within any of the user's nodes.
  - Valid node descriptions consist of:
    - minimum of 1 character and are limited to 300 characters
- User selects valid tags
  - Valid tags are found from a bank of tags provided
  - A created node does not require a tag
- User selects valid privacy mode
  - Privacy toggles between private and public.
    - By default a node will inherit it's parent privacy mode.
    - A child cannot be public if it's parent is private.
- A maximum of ____ nodes can be created or pasted within 24 hours with the same account before a hard limit is reached and node creation is disabled.
- A maximum of ____ nodes can be created or pasted before a user reaches a soft limit.

Specific Input Example:

Name: Octopush

Description: Contact sport in which two teams attempt to move a puck into the opposing team's goal. The sport takes place on the bottom of a swimming pool.

Privacy: Public

Tags: Hobbies

Test Steps:

1. User clicks the parent node
2. User clicks *Add Node* on the context menu
3. If the user has reached soft limit, user will
    a. Solve captcha
    b. Click *OK*
4. User enters node name into *Name* input field
5. User enters node description into *Description* input field
6. User toggles between privacy modes desired.
7. If the user desires to add tags
    a. User clicks *add tags*
    b. Search for relevant tags in search bar
    c. Click on tag to add
    d. Click *Close*
8. User clicks *Save* button

Success Cases:

- User creates and saves a node with valid inputs and parent. The user is automatically navigated to a page containing an overview of the tree that contains the created node with the appropriate parent .
- User creates and saves a root node with valid inputs and parent . User meets the soft limit and successfully solves captcha. The user is automatically navigated to a page containing an overview of the tree that contains the created node with appropriate parent.

Failure Cases:

- User creates and saves a node with valid inputs and parent. The automatic navigation does not take place.
- User creates and saves a root node with valid inputs and parent. User is navigated to a page other than the new tree's overview.
- User creates and saves a root node with valid inputs. Database connection is broken. System message displays "This feature is currently undergoing maintenance. Please try again at a later time."
- User creates and saves a root node with empty inputs and valid parents. System message displays "Input field cannot be left empty."
- User creates a root node with an invalid parent (parent has no room for children). Context menu does not allow "Add child" to be clicked.
- User attempts to create a node after the user's hard limit has been reached. System message displays "You have reached your limit for nodes today. The node was not created. Please delete some nodes or try again tomorrow."

- User creates and saves a node with a name that already exists in the user's portal. System message displays "There is already a node with that title."

Non-functional Requirements:

- New page containing node's input fields is loaded within five seconds
- User tree database is updated within 5 seconds
- User's hard limit count is incremented by one
- User's soft limit count is incremented by one
- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.3 Copying A Single Node (Estimated 2.5 Hours)

Purpose:

Verify that a user can copy a single node into a user's clipboard.

Preconditions:

- User is active authenticated user
- User has a stable internet connection
- User has navigated to another user's tree portal page with at least one node (the node copied cannot be owned by the user)

Test Steps:

1. Click on the node to be copied
2. Click on *Copy Node* on the context node

Success Cases:

- User copies a node. System message displays "Node copied."Node information is copied onto the user's clipboard. This includes
  - Node Name
  - Node Description
  - Node Tags
  - Node Privacy Settings

Failure Cases:

- User copies a node. The System message does not display.
- User copies a node. The page is navigated to any page other than the current page.
- User copies a node. Database connection is broken. System message displays "This feature is currently undergoing maintenance. Please try again at a later time."

Non-functional Requirements:

- Copy data will be updated within 5 seconds.
- Function will be accessible 90% of the time.
- Time to repair the function will be at least 1 hour and within 24 hours on average.

9.4 Copying Multiple Nodes (Estimated 2.5 Hours)

Purpose:

Verify that a user can copy multiple nodes into a user's clipboard.

Preconditions:

- User is active authenticated user
- User has a stable internet connection
- User has navigated to another user's tree portal page with multiple nodes (the nodes copied cannot be owned by the user)

Test Steps:

1. Shift + click nodes in the same branch that are to be copied.
2. Click on *Copy Nodes* on the context menu

Success Cases:

- User copies multiple nodes. Nodes that are copied are now indicated with a blue hue. System message displays "Nodes copied." Nodes' information is copied onto the user's clipboard. This includes
  - Node Name
  - Node Description
  - Node Tags
  - Node Privacy Settings

Failure Cases:

- User copies multiple nodes. The System message does not display.
- User copies multiple nodes. The page is navigated to any page other than the current page.
- User copies multiple modes that are *not* connected (different subtrees). System message displays "The nodes copied are not connected. Retry with connected nodes."
- User copies multiple nodes. Database connection is broken. System message displays "This feature is currently undergoing maintenance. Please try again at a later time."

Non-functional Requirements:

- Copy data will be updated within 5 seconds.
- Function will be accessible 90% of the time.
- Time to repair the function will be at least 1 hour and within 24 hours on average.

9.5 Pasting node(s) into a New Tree (Estimated 2.5 Hours)

Purpose:

Verify that a user can paste node(s) from a user's clipboard into a new tree on the user's tree portal

Requirements:

- A maximum of ____ nodes can be created or pasted within 24 hours with the same account before a hard limit is reached and node creation is disabled.
- A maximum of ____ nodes can be created or pasted before a user reaches a soft limit.

Preconditions:

- User is active and authenticated user
- User has a stable internet connection
- User has navigated to user's tree portal page
- User has node(s) copied in their clipboard

Test Steps:

1. Click on empty space on the user's tree portal page
2. Click *Paste Node(s)* on the context menu
3. If User has reached soft limit, user will
    a. Solve captcha
    b. Click *OK*
4. Click the *Save* Button

Success Cases:

- User pastes nodes onto the tree page. The user is automatically navigated to a page that contains an overview of the new tree.
- User pastes nodes onto the tree page. The user hits the soft limit and successfully solves the captcha. The user is automatically navigated to a page that contains an overview of the new tree.

Failure Cases:

- User pastes node(s). The user is not automatically navigated to a page that contains an overview of the new tree.
- User pastes node(s). The user is automatically navigated to any other page except the page that contains an overview of the new tree.
- User attempts to paste a node(s) after the user's hard limit has been reached. System message displays "You have reached your limit for nodes today. The node(s) were not created. Please delete some nodes or try again tomorrow."
- User attempts to paste multiple nodes and the user's hard limit has been reached while pasting. System message displays "You have reached your limit for nodes today. The nodes were not created. Please delete some nodes or try again tomorrow."

Non-functional Requirements:

- Tree database will be updated within 5 seconds.
- Function will be accessible 90% of the time.
- Time to repair the function will be at least 1 hour and within 24 hours on average.

9.6 Pasting node(s) into an Existing Tree (Estimated 2.5 hours)

Purpose:

Verify that a user can paste node(s) into an existing tree

Preconditions:

- User is an active and authenticated user
- User has a stable internet connection
- User has navigated to user's tree portal page
- User's tree portal page contains at least one node

Requirements:

- A maximum of ____ nodes can be created or pasted within 24 hours with the same account before a hard limit is reached and node creation is disabled.
- A maximum of ____ nodes can be created or pasted before a user reaches a soft limit.

Test Steps:

1. Click on the node who will be the parent node
2. Click on *Paste Node(s)* on the context menu
3. If user has reached soft limit, user will
    a. Solve captcha
    b. Click *Ok* button
4. Click the *save* button

Success Cases:

- User pastes node(s) onto the parent node. The user is automatically navigated to a page that contains an overview of the updated tree.
- User pastes node(s) onto the parent node. The user hits the soft limit and successfully solves the captcha. The user is automatically navigated to a page that contains an overview of the updated tree.

Fail Cases

- User pastes node(s) onto the parent node. The user is not automatically navigated to a page that contains an overview of the new tree
- User pastes node(s) onto the parent node. The user is automatically navigated to any other page except the page that contains an overview of the updated tree.
- User attempts to paste node(s) after the user's hard limit has been reached. System message displays "You have reached your limit for nodes today. The node(s) were not created. Please delete some nodes or try again tomorrow."

Non-functional Requirements

- Tree database will be updated within 5 seconds
- Function will be accessible 90% of the time
- Time to repair the function will be at least 1 hour and within 24 hours on average

9.7 Setting node(s) private/public settings (Estimated 3 hours)

Purpose

Verify that a user can toggle the privacy settings between private and public

Preconditions

- User is an active and authenticated user
- User has a stable internet connection
- User has navigated to the user's tree portal page
- User's tree portal page contains an existing tree with at least one node

Requirements

- User selects a valid privacy node
    - Privacy toggles between private and public
        - By default a node will inherit it's parent privacy mode
        - A child cannot be public if it's parent is private
- User selects multiple nodes to change privacy
    - The nodes must exist within the same subtree
    - The initial privacy mode for the selected nodes will be the node with the lowest height
    - The nodes cannot be public if the node with the lowest height's parent is private
- A child cannot be public if it's parent is private
    - The user will not be able to toggle the node(s) if it's parent is private

Specific Input

- Privacy is toggled private
- Privacy is toggled public

Test Steps

1. User shift + clicks on node(s)
2. User clicks on *Edit Privacy* in the context menu
3. User toggle clicks between privacy states
4. User clicks *Ok*
5. User clicks *Save*

Success Case

- User toggles the privacy setting of a node. The user is automatically navigated to a page containing an overview of the tree containing the updated nodes.

Fail Case

- User toggles the privacy setting of a node. The automatic navigation does not take place
- User toggles the privacy setting of a node. The user is automatically navigated to a page other than the overview of the tree containing the updated nodes.

Non-functional Requirements

- Overview of the tree containing the updated nodes is loaded within five seconds
- User tree database is updated within 5 seconds
- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.8 Changing contents of a node (Estimated 2 hours)

Purpose

Verify that a user can change the contents of a node.

Preconditions

- User is an active and authenticated user
- User has a stable internet connection
- User has navigated to the user's tree portal with an existing tree

Requirements

- User enters valid name and descriptions in relative input fields
    - Valid node names consist of:
        - minimum of 1 character and are limited to 50 characters.
        - Unique. The name must not already exist within any of the user's nodes.
    - Valid node descriptions consist of:
        - minimum of 1 character and are limited to 300 characters
- User selects valid tags
    - Valid tags are found from a bank of tags provided
    - A created node does not require a tag
- User selects valid privacy mode
    - Privacy toggles between private and public.
        - By default a node will inherit it's parent privacy mode.
        - A child cannot be public if it's parent is private.

Specific Input Example:

Name: The Art of Walking

Description: Figuring out how to move in a fast paced world.

Privacy: Private

Tags: Hard

Test Steps

1. Click on node
2. Click on *Edit Node* on the context menu
3. User updates node name in the *Name* input field
4. User updates node description in the *description* input field
5. User toggles between privacy modes desired
6. If the user desires to add tags
    a. User clicks *add tags*
    b. Search for relevant tags in search bar
    c. Click on tag to add
    d. Click *Close*
7. User clicks *Save* button

Success Cases

- User updates and saves nodes with valid inputs. The user is automatically navigated to a page containing an overview of the tree with an updated node.

Fail Cases

- User updates and saves nodes with valid inputs. The user automatic navigation does not take place

- User updates and saves nodes with valid inputs. The user is automatically navigated to a page other than the tree's overview
- User updates and saves nodes with valid inputs. The user is automatically navigated to a page containing an overview of the tree but the node is not updated.

Non-functional Requirements

- Overview of tree with updated node is loaded within five seconds
- User tree database is updated within 5 seconds
- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.9 Adding a tag to a node (Estimated 1 hour)

a. Test Case Description
    i. Verify that a user can add a tag to an existing node that the user owns
b. Preconditions
    i. User is logged on with a valid username and password
    ii. User has internet connection
    iii. User is on user's tree portal page
    iv. User's tree portal page contains an existing tree with at least one node
c. Test Steps
    i. Click on node (or shift + click multiple nodes)
    ii. Click on *Edit Tag(s)*
    iii. Search for relevant tags in search bar
    iv. Click on tag to add
    v. Click *Close*
    vi. Click *Save*
d. Specific Input Example
    i. Tag Names: Math, UX, Discord
e. Expected Result
    i. Tag is added to database
    ii. Tag pops up on side of node view
    iii. Tag pops up with other existing tags when editing tags
f. Fail Cases
    i. Tag does not exist
        1. Error message explaining tag does not exist in search bar
        2. User can continue searching in search bar
    ii. Loss of internet connection
        1. Error message explaining loss of internet connection
        2. No changes made in database
        3. User will be directed to user portal
    iii. No connection made to cloud server
        1. Error message explaining loss of connection to cloud server
        2. User will be directed to user portal
g. Non-functional requirements
    i. User tree data updated within 5 seconds
    ii. User tree page visually updated within 5 seconds
    iii. Function will be accessible 90% of the time
    iv. Time to repair function will be at least 1 hour and within 24 hours on average

9.10 Deleting a tag from a node (Estimated 1 hour)

a. Test Case Description
    i. Verify that a user can remove a tag to an existing node that the user owns
b. Preconditions
    i. User is logged on with a valid username and password
    ii. User has internet connection
    iii. User is on user's tree portal page
    iv. User's tree portal page contains an existing tree with at least one node
    v. User's node has at least one tag to remove
c. Test Steps
    i. Click on node (or shift + click multiple nodes)
    ii. Click on *Edit Tag(s)*
    iii. Click on the *x* on the tag to remove
    iv. Click *Close*
    v. Click *Save*
d. Specific Input Example
    i. None
e. Expected Result
    i. Tag is removed from database
    ii. Tag no longer appears on nodes view
    iii. Tag no longer appears on existing tags when editing tags
f. Fail Cases
    i. Loss of internet connection
        1. Error message explaining loss of internet connection
        2. No changes made in database
        3. User will be directed to user portal
    ii. No connection made to cloud server
        1. Error message explaining loss of connection to cloud server
        2. User will be directed to user portal
g. Non-functional requirements
    i. User tree data updated within 5 seconds
    ii. User tree page visually updated within 5 seconds
    iii. Function will be accessible 90% of the time
    iv. Time to repair function will be at least 1 hour and within 24 hours on

9.11 Changing a parent of a node(s) (Estimated 6 hours)

a. Test Case Description
    i. Verify that the user can change the parent of a node
b. Preconditions
    i. User is logged on with a valid username and password
    ii. User has internet connection
    iii. User is on user's tree portal page
    iv. User's tree portal page contains an existing tree with at least two nodes
c. Test Steps
    i. Click on node
    ii. Click on *Change Parent*
    iii. Nodes that can be used will be highlighted
    iv. Click new parent to be used
    v. Click *Save* button
d. Specific Input Example
    i. None
e. Expected Result
    i. Database will update node's parent
    ii. User will see that the node's parent has been changed
f. Fail Cases
    i. Loss of internet connection
        1. Error message explaining loss of internet connection
        2. No changes made in database
        3. User will be directed to user portal
    ii. No connection made to cloud server
        1. Error message explaining loss of connection to cloud server
        2. User will be directed to user portal
g. Non-functional requirements
    i. User tree data updated within 5 seconds
    ii. User tree page visually updated within 5 seconds
    iii. Function will be accessible 90% of the time
    iv. Time to repair function will be at least 1 hour and within 24 hours on

9.12 Deleting a Single Node (Estimated 2.5 hours)

a. Test Case Description
    i. Verify that a user can delete a single node
b. Preconditions
    i. User is logged on with a valid username and password
    ii. User has internet connection
    iii. User is on user's tree portal page
    iv. User's tree portal page contains an existing tree with at least one node
c. Test Steps
    i. Click on node
    ii. Click *Delete Node*
    iii. Click Save Button
d. Specific Input Example
    i. None
e. Expected Result
    i. Node is removed from database
    ii. Node no longer appears on user's tree portal page
f. Fail Cases
    i. Deleting a parent of a dangling node
        1. Error message explaining that node cannot be deleted without deleting child
        2. No changes made in database
        3. User will be redirected to user portal
    ii. Loss of internet connection
        1. Error message explaining loss of internet connection
        2. No changes made in database
        3. User will be directed to user portal
    iii. No connection made to cloud server
        1. Error message explaining loss of connection to cloud server
        2. User will be directed to user portal
g. Non-functional requirements
    i. User tree data updated within 5 seconds
    ii. User tree page visually updated within 5 seconds
    iii. Function will be accessible 90% of the time
    iv. Time to repair function will be at least 1 hour and within 24 hours on

9.13 Deleting Multiple Nodes (Estimated 2.5 Hours)

a. Test Case Description
    i. Verify that a user can delete multiple nodes at once
b. Preconditions
    i. User is logged on with a valid username and password
    ii. User has internet connection
    iii. User is on user's tree portal page
    iv. User's tree portal page contains an existing tree with at least two nodes
c. Test Steps
    i. Shift + click on node
    ii. Click *Delete Nodes*
    iii. Click Save Button
d. Specific Input Example
    i. None
e. Expected Result
    i. Nodes are removed from database
    ii. Nodes no longer appears on user's tree portal page
f. Fail Cases
    i. Deleting a parent of a dangling node
        1. Error message explaining that node cannot be deleted without deleting child
        2. No changes made in database
        3. User will be redirected to user portal
    ii. Loss of internet connection
        1. Error message explaining loss of internet connection
        2. No changes made in database
        3. User will be directed to user portal
    iii. No connection made to cloud server
        1. Error message explaining loss of connection to cloud server
        2. User will be directed to user portal
g. Non-functional requirements
    i. User tree data updated within 5 seconds
    ii. User tree page visually updated within 5 seconds
    iii. Function will be accessible 90% of the time
    iv. Time to repair function will be at least 1 hour and within 24 hours on

9.14 Searching for a topic without any filters (Estimated 2 hours)

Purpose:

Verify that a user can search for node topics

Preconditions:

- User has an active and authenticated user
- User has a stable internet connection
-

a. Test Case Description
   i. Verify that a user can search for node topics
b. Preconditions
   i. User has internet connection
   ii. User is using search bar (either on search page or search bar on homepage)
c. Test Steps
   i. User enters topic to search
   ii. User presses *Enter*
d. Specific Input Example
   i. Search Example: UX, Cooking
e. Expected Result
   i. User is directed to page listing of all nodes that contains a match to topic
   ii. If there are no resulting matches, user is notified
f. Fail Cases
   i. Loss of internet connection
      1. Error message explaining loss of internet connection
      2. No changes made in database
      3. User will be directed to user portal
   ii. No connection made to cloud server
      1. Error message explaining loss of connection to cloud server
      2. User will be directed to user portal
g. Non-functional requirements
   i. User tree data updated within 5 seconds
   ii. User tree page visually updated within 5 seconds
   iii. Function will be accessible 90% of the time
   iv. Time to repair function will be at least 1 hour and within 24 hours on

9.15 Applying filters to the results of a search (Estimated 1.5 hours)

Purpose

Verify the user can add filters to the results of a search

Preconditions:

- User is an active and authenticated user
- User has a stable internet connection
- User has navigated to the search page

Specific Input Example:

Topic: Extreme Unicycling
Tags: Hard

Test Steps:

1. User searches for word to add to filter
2. User clicks on word bank to add to filter
3. User can switch between radio buttons to sort by relevance or sort by rating
4. User clicks *Filter*

Success Case:

- User searches for topics and filters. User is automatically navigated to a page containing search results that contain the topic and filter.

Failure Case:

- User searches for topics and filters. The automatic navigation does not take place.
- User searches for topics and filters. User is navigated to a page other than the search results that contain the topic and filter.
- User searches for topics and filters. User is automatically navigated to a page continuing search results but is not filtered

Non-functional Request:

- Page containing search results is loaded within five seconds
- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

a. Test Case Description
   i. Verify the user can add filters to the results of a search
b. Preconditions
   i. User has internet connection
   ii. User has already searched a topic
   iii. User is on the search results page
c. Test Steps
   i. User searches for word to add to filter
   ii. User Clicks on word bank to add to filter
   iii. User can switch between radio buttons to sort by relevance or sort by rating
   iv. User clicks *Filter*
d. Specific Input Example
   i. Word bank search: cooking
e. Expected Result
   i. User is directed to page listing of all nodes that contains a match to topic
   ii. If there are no resulting matches, user is notified
f. Fail Cases
   i. Loss of internet connection
      1. Error message explaining loss of internet connection

        2. No changes made in database
        3. User will be directed to user portal
    ii. No connection made to cloud server
        1. Error message explaining loss of connection to cloud server
        2. User will be directed to user portal
g. Non-functional requirements
    i. User tree data updated within 5 seconds
    ii. User tree page visually updated within 5 seconds
    iii. Function will be accessible 90% of the time
    iv. Time to repair function will be at least 1 hour and within 24 hours on

9.16 Applying filters before searching for a topic (Estimated 1.5 Hours)

Purpose:

Verify the user can add filters before searching for a topic.

Preconditions:

- User is an active and authenticated user
- User has a stable internet connection
- User has navigated to the search page

Specific Input

Tags: Hard

Topic: History of Soap Operas

Test Steps:

1. User searches for tag to add to filter
2. User clicks on word in word bank to add filter
3. Usr can switch between radio buttons to sort by relevance or sort by rating
4. User enters topic in search bar
5. User clicks *Enter*

Success Case:

- User filters and searches a topic. User is automatically navigated to a page containing search results that contain the topic and filter.

Failure Cases:

- User filters and searches a topic. The automatic navigation does not take place.
- User filters and searches a topic. User is navigated to a page other than the search results that contain the topic and filter.
- User filters and searches a topic. User is automatically navigated to a page continuing search results but is not filtered

Non-functional Requirements

- Page containing search results is loaded within five seconds
- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.17 Rating a Single Node (Estimated 1 Hour)

Purpose:

Verify that a user can rate another user's node

Preconditions

- User is active and authenticated user
- User has a stable internet connection
- User has navigated to another user's tree portal page with at least one node

Requirements

- A user selects a valid rating
    - Valid rating consist of
        - A rating between 1 and 5

Specific Input Example

- User rates another user's node at 2 stars

Test Steps

1. User clicks node
2. User clicks *Rate Node* on the context menu
3. User clicks on the rating to give
4. User clicks *close* button

Success Case

- User rates node. User is automatically navigated to the node's page with an updated score

Failure Case

- User rates node. The automatic navigation does not take place.
- User rates node and is navigated to a page other than the node's page with an updated score.
- User rates node and is navigated to the node's page that does not have an updated score.

Non-functional Requirements

- Page containing node with updated score is loaded within 5 seconds
- Node's rating database is updated within five seconds
- Function will be accessible 90$ of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.18 Rating Multiple Nodes (Estimated 1 Hour)

Purpose:

Verify that a user can rate multiple nodes of another users

Preconditions:

- User is active and authenticated user
- User has a stable internet connection
- User has navigated to a user's tree portal with an existing tree and at least two nodes

Requirements

- A user selects a valid rating
    - Valid rating consist of
        - A rating between 1 and 5

Specific Input Example

- User rates another user's nodes at 2 stars

Test Steps

1. User shift + clicks node
2. Usr clicks *Rate Nodes* on context menu
3. User clicks on rating to give
4. Usr clicks *close*

Success Case

- User rates nodes. User is automatically navigated to the tree page with an updated score

Failure Case

- User rates nodes. The automatic navigation does not take place.
- User rates nodes and is navigated to a page other than the tree's page with an updated score for the nodes.
- User rates nodes and is navigated to the tree's page that does not have an updated score.

Non-functional Requirements

- Page containing nodes with updated score is loaded within 5 seconds
- Nodes' rating database is updated within five seconds
- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.19 User Authentication

Purpose:

The purpose of this use case is for the user to authenticate themself as a valid user of the system.

Scope:

This Use Case deals with Users attempting to use the system

Actors:

- User
- System Administrator

Requirements:

- A valid user is defined as a user that has been authenticated by the system with valid security credentials
- Valid security credentials consist of a username and a password
    - Valid usernames will consist of the following
        - a-z
        - 0-9
        - .,@!
    - Valid security credentials will be a valid time-based one-time password (OTP) as defined in NIST SP 800-63b section 5.1.4.1. Valid passwords will be defined as follows
        - Password must be at minimum 8 characters and with a maximum of 100 characters
        - OTP passwords expire after 2 minutes
        - OTP passwords are changed after reach use
        - Valid characters will include
            - a-z
            - A-Z
            - 0-9
            - ,.@!
- Five failed authentication attempts within 24 hours will result in the account being disabled
    - The first failed authentication attempt will be the start of this 24 hour period. Once the 24 hour period is up, the number of failed attempts will be reset to zero.
    - Every failed authentication attempt will log the account attempted to be signed in and the IP address used

Pre-Conditions:

- User must **not** be an active and authenticated user otherwise the user can not attempt to authorize.
- User must navigate to the login page.
- User must have access to valid security credentials

Test Steps:

1. Enter username in *Username* input field
2. Enter password in *Password* input field
3. Click *Login*

Success Conditions:

- User submits valid security credentials and is automatically navigated to the user's tree page

Failure Conditions:

- User submits valid security credentials and is not automatically navigated to any page.
- User submits valid security credentials and is automatically navigated to a page other than the user's tree page.
- User submits invalid username. A system message displays "Invalid username or password. Try again." The failed attempt count increases by one.
- User submits invalid password. A system message displays "Invalid username or password. Try again." The failed attempt count increases by one.
- User submits invalid username or invalid password. A system message displays "Invalid username or password. Try again.". The failed attempt count does not increase by one.
- User submits valid security for a disabled account. A system message displays "Account disabled. Contact system admin." Failed attempt count is increased by one.
- User submits valid security for a disabled account. A system message displays "Account disabled. Contact system admin." Failed attempt count is not increased by one.
- User submits valid security for a disabled account. A system message displays "Account disabled. Contact system admin." IP Address and account is not logged.

Non-Functional Requirements:

- Users redirection to tree overview is loaded within five seconds
- Database containing user's failed attempt count is updated within five seconds
- Database containing log is updated within five seconds
- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.20 Account Creation (Registration)

Purpose:

Mechanism for creating new user accounts within the system.

Scope:

Any user attempting to use the system.

Actors:

- User

Requirements:

- System administrators can not be created by this feature
- All user account information will be stored in a persistent data store
- Valid security credentials consist of a username and a password
    - The users username will be a valid email and will consist of the following
        - a-z
        - 0-9
        - .,@!
    - Valid passwords will be defined as follows
        - Password must be at minimum 8 characters and with a maximum of 100 characters
        - Valid characters will include
            - a-z
            - A-Z
            - 0-9
            - ,.@!
- The user must provide their first and last name
    - Valid first and last names will consist of the following
        - a-z
        - A-Z
        - -,
    - User must certify that the user is at least 13 years of age
        - The user must be at least 13 years of age to user our system in order to comply with the Children's Online Privacy Protection Act (COPPA)

Preconditions:

- User must not be an active and authorized user
- User has access to a valid email
- User has navigated to the sign up page

Test Case:

1. User enters valid first name in the *first name* input field
2. User enters valid last name in the *last name* input field
3. User enters valid email in the *email* input field
4. User enters valid password in the *password* input field
5. User clicks on the policy agreement checkbox
6. User clicks *Create Account* button

Success Condition:

- User registers with a valid email and valid password. User is automatically directed to a new tree overview view.

Failure Condition:

- User registers with a valid email and password. User is not automatically directed to any page and displays no system message
- User registers with a valid email and password. User is automatically directed to a page other than the new tree overview view.
- User registers with an invalid email. System displays “Unable to create account. Invalid username.”
- User registers with an invalid password. System displays “Unable to create account. Invalid password. Please use a-z, A-Z, 0-9 and ,.@!”

Non-Functional Requirements:

- Users redirection to tree overview is loaded within five seconds
- Database containing accounts is updated within five seconds
- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.21 Logout

Purpose:

The purpose of this mechanism is to end a session for an active and authorized user

Scope:

Any active and authorized user attempting to use the system

Requirements:

- User's sessions ends within five seconds on logout request
    - User's end of session will redirect user to homepage

Preconditions:

- User is an active and authenticated user
- User has navigated to to the users tree page overview

Test Case:

1. User clicks *logout* button

Success Conditions:

- User clicks logout and ends the active session. The user is automatically navigated to the homepage with the options to log in /sign up at the top right.

Failure Conditions:

- User clicks logout and ends the active session. The automatic navigation does not take place
- User clicks logout and ends the active session. The user is automatically navigated to a page other than the homepage.
- User clicks logout and ends the active session. User is automatically navigated to the homepage but there is no login/sign up option.
- The logout request takes longer than 5 seconds

Non-Functional Requirements:

- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.21 Authorization (Estimated 3 Hours)

Purpose:

The purpose of authentication is to identify if a user is a valid user of the system.

Scope:

The scope of this component covers all users that are attempting to use the system.

Requirements:

- Unauthenticated users will only be given access to resources that do not require knowledge of the user's identity.
- Unauthorized users will not be able to view, modify or delete any protected data
- Unauthorized can not view or interact with protected views or functions
- The operation and timestamp of each unauthorized access will be recorded
- Users access modifications will be restored with unauthorized user is authenticated

Preconditions:

- User must is an active and authorized user
- User is attempting to access a view that is not authorized

Test Case:

1. User clicks on unauthorized page

Success Condition:

- User attempts to use a protected feature within authorization scope and access is granted
- User attempts to view protected data within authorization scope and access is granted
- User attempts to modify protected data within authorization scope and access is granted
- User attempts to to view protected view within authorization scope and access is granted

Failure Condition:

- User attempts to use a protected feature outside authorization scope and access is denied
- User attempts to view protected data outside authorization scope and access is denied
- User attempts to modify protected data outside authorization scope and access is denied
- User attempts to to view protected view outside authorization scope and access is denied

Non-Functional Requirements:

- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.22 Deletion (Estimated 2 Hours)

Purpose:

The purpose of this mechanism is to delete a user account

Scope:

Any registered user of the system

Requirements:

- System administrators are only able to be deleted by another system administrator
- All personal identifiable information (PII) along with user's account data is deleted from the system permanently
    - This deletion is irreversible

Preconditions:

- User must be an active and authenticated user
- User be navigated to the delete account view
- User has permission to delete the account. See system administrator permission in requirements.

Test Case:

1. User clicks *Delete Account*
2. User clicks *Yes*

Success Conditions:

- User deletes account and confirms deletion. All personal identifiable information is deleted and a System message displays "Account deleted." User is automatically navigated to the homepage

Failure Conditions:

- User deletes account and confirms deletion. Personal identifiable information is not deleted.
- User deletes account and confirms deletion. Automatic navigation does not take place
- User deletes account and confirms deletion. User is automatically navigated to a page other than the homepage.
- User deletes account and confirms deletion. System message does not display.

Non-Functional Requirements:

- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.23 User Account Management (Estimated 5 Hours)

Purpose:

The purpose of having a User Account Management feature is to provide a mechanism for system administrators to observe and make changes to accounts if necessary

Scope:

The scope of this component covers all users that have a registered account within our system.

Requirements:

- Operations will be applied to a persistent data store.
- Users must be a system administrator to access this feature
- The system administrator has access to view and modify accounts and its corresponding data within the entire system
- System administrators can perform single operations as follows
    - Create Account
    - Update Account
    - Delete Account
    - Disable Account
    - Enable Account
    - Single Operations will be completed within 5 seconds
- System administrators can perform bulk operations
    - Bulk operations can consist of multiple operations of the same type or mixed type
    - A maximum of 10,000 operations per request
    - Requests can be made from an uploaded file extract
        - The uploaded file must be less than 2 GB in size
    - Bulk operations must be completed within 60 seconds
- Single and bulk operations will affect all users and their attributes within the system
- System administrators can create new system administrator accounts
- At any given type, the system must have at least one system administrator

Preconditions:

- User must be an active and authenticated user
- User must have system administration privileges
- User must be navigated to the user management view

Success Conditions:

- User performs a single user management operation within 5 seconds upon request. System message displays “Single operation successful.”
- User performs bulk operations containing less than 10,000 operations within 60 seconds. System message displays “Bulk operation successful.”
- User performs bulk operations containing 10,000 operations within 60 seconds. System message displays. “Bulk operation successful.”

Failure Conditions:

- User's single operation takes more than 5 seconds
- User's bulk operation takes more than 60 seconds
- 10k bulk operation takes more than 60 seconds
- Single operation takes five seconds but no system message is displayed
- Single operation takes five seconds but wrong system message is displayed
- Bulk operation takes 60 seconds but no system message is displayed
- Bulk operation takes 60 seconds but wrong system message is displayed
- 10k bulk operation takes 60 seconds but no system message is displayed
- 10k bulk operation takes 60 seconds but wrong system message is displayed
- Single operation takes five seconds and displays system message but data isn't stored in persistent database
- Bulk operation takes 60 seconds and displays system message but data isn't stored in persistent database
- 10k bulk operation takes 60 seconds and displays system message but data isn't stored

Non-Functional Requirements:

- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.24 Usage Analysis Dashboard (Estimated 6 Hours)

Purpose:

The purpose of this mechanism is to provide a visualization on user behavior in the system

Scope:

The scope of this component covers system administrators account of the system

Requirements:

- All data is fetched from a persistent data store
- Usage analysis dashboard contains the following key performance indicators (KPI)
    - A bar chart containing the top five most most visited views of all time
    - A bar chart containing the top five average duration per view of all time
    - A trend chart of the number of logins per day within a 3 month span
    - A trend chart of the number of registrations per day within a 3 month span
    - A bar chart containing the top five most searched topics within a week
    - A bar chart containing the top five most tagged topics within a week
- All key performance indicators are refreshed automatically every sixty seconds
- Usage analysis dashboard view must be loaded within 15 seconds

Preconditions:

- User is an active and authenticated user
- User is a system administrator
- User has a stable internet connection
- User has navigated to to the usage analysis dashboard

Success Conditions:

- User is able to navigate to the usage analysis dashboard view. The view is loaded within 15 seconds and refreshes its data within 60 seconds

Failure Conditions:

- User is unable to navigate to the usage analysis dashboard view but is a system administrator
- User is able to navigate to the usage analysis dashboard view but doesn’t load within 15 seconds
- User is able to navigate to the usage analysis dashboard view within 15 seconds but the key performance indicators data is not refreshed every 60 seconds

- User is able to navigate to the usage analysis dashboard view within 15 seconds but not all of the key performance indicator data is refreshed within 60 seconds
- User is able to navigate to the usage analysis dashboard view within 15 seconds

Non-Functional Requirements:

- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.25 Logging (Estimated 5 Hours)

Purpose:

The purpose of this mechanism is to provide event tracking for system audits

Scope:

The scope of this component covers system events and user events.

Functional Requirements:

- Log entries are immutable (unable to be changed)
- Log entries will be stored in a persistent database
- All log entries will consist of
    - UTC timestamp of occurrence
    - Log Level
    - User who performed operation
    - Category of log
    - Description of log
- Valid log levels include
    - Info - tracks the flow of the system
    - Debug - information to help maintainers of the system
    - Warning - events that can show system failures
    - Error - shows system errors
- Valid categories
    - View
    - Business
    - Server
    - Data
    - Data store
- Logging process must not interfere or block interactions within the system
- Logging process must not take more than five seconds.

Preconditions:

- Data is stored within an active persistent data store
- Persistent data store must be accessible by the system
- Persistent data store must have storage capacity for a new log entry

Tests:

- Xunit unit tests to include tests of each method contained in
    - Log.cs
    - LogService.cs

Success Condition:

- System logs successful user events within five seconds
- System logs failed user events within five seconds
- System logs successful user events within five seconds
- System logs failed user events within five seconds

Failure Condition:

- System log process takes longer than five seconds
- System log process interferes or blocks interactions within the system
- System log doesn't store to a persistent database
- System log data store doesn't have the capacity to store a new long entry
- System log event is able to be modified
- System log event is stored with inaccurate entry

Non-Functional Requirements:

- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average

9.26 Archiving (Estimated 2 Hours)

Purpose:

The purpose of this mechanism is to offload log entries to preserve system resources

Scope:

The scope of this component covers all log entries made by the system

Functional Requirements:

- On every 1st of the month at 00:00:00AM (local time) the archival process will execute
- Only log entries older than 30 days will be offloaded
- Archived entries will be consolidated and compressed
- Offloaded log entries will be moved to another location
- Offloaded log entries will be removed after successful archival
- Log entries archival process must be completed within 60 seconds of request

Preconditions:

- On every 1st of the month at 00:00:00AM (local time) the archival process will execute
- Only log entries older than 30 days will be offloaded
- Archived entries will be consolidated and compressed
- Offloaded log entries will be moved to another location
- Offloaded log entries will be removed after successful archival
- Log entries archival process must be completed within 60 seconds of request

Test Case:

1. Click *Archive Logs*

Success Condition:

- Archival process begins at 00:00:00AM (local time) on the 1st of each month. Logs older than 30 days are offloaded. Each offloaded log is consolidated, compressed, moved to another location and removed within 60 seconds.

Failure Condition:

- Archival process did not start at 00:00:00AM
- Archival process did not begin at 00:00:00AM local time
- Archival process began at 00:00:00AM local time but not on the first of the month
- Archival process began at 00:00:00AM local time on the first of the month but logs older than 30 days are not offloaded
- Archival process began at 00:00:00AM local time on the first of the month. Logs older than 30 days are offloaded but did not offload all log entries older than 30 days.
- Archival process began at 00:00:00AM local time on the first of the month. All log entries older than 30 days are offloaded but not consolidated.

- Archival process began at 00:00:00AM local time on the first of the month. All log entries older than 30 days are offloaded and consolidated. Log entries are not compressed
- Archival process began at 00:00:00AM local time on the first of the month. All log entries older than 30 days are offloaded, consolidated and compressed but not moved to another location.
- Archival process began at 00:00:00AM local time on the first of the month. All log entries older than 30 days are offloaded, consolidated and compressed. Log entries are moved to another location but not removed
- Archival process began at 00:00:00AM local time on the first of the month. All log entries older than 30 days are offloaded, consolidated and compressed. Log entries are moved to another location and not removed.
- Archival process began at 00:00:00AM local time on the first of the month. All log entries older than 30 days are offloaded, consolidated and compressed. Log entries are moved to another location and removed. The process takes longer than 60 seconds.

Non-Functional Requirements:

- Function will be accessible 90% of the time
- Time to repair function will be at least 1 hour and within 24 hours on average