

BRD

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
10/04/2021

California State University, Long Beach
College of Engineering
CECS491 Section 6 11950, Fall 2021

Scope

The project will be a multiple page web application. We will support the latest version of Chrome at the date that we start development (we will maintain our Chrome versions at this version or downgrade the presenters version to it when necessary). Our initial target audience will be American English speaking residents of the United States above the age of 15 to comply with [Children's Online Privacy Protection Act \(COPPA\)](#). The IP range will be restricted to the United States to maintain our target customers. The web application will utilize cookies in order to enhance the user experience and for any other functionalities that may be enhanced through their use that we choose to apply the cookies towards. Cookies will adhere to the strongest regional regulations within the United States.

Use Case 1: Creating a node

Actors: User

Functions:

1. User left clicks on create tree button
2. User fills input fields
 - a. Title field
 - i. Limited to #### characters
 - b. Description field
 - i. Limited to #### characters
3. User selects tags
4. Toggle inherit tags button
5. Toggle private/public button
6. User clicks confirm button

Variations:

1. User left clicks existing node
 - a. User left clicks on create child button
 - b. Same process as default
 - c. Branch is automatically made to connect parent node to child node
 - d. Child inherits parent private/public state
 - e. If parent is public, child can be made private during creation

Post Conditions:

User tree data updated with data of the created node

User soft limit count incremented by one

If within 5 of soft limit, bring up soft limit message

- Nodes left before soft limit reached

If soft limit reached

- Message for soft limit reached
- Trigger recaptcha
- Message for how many soft limits left, how many nodes left before hard limit

User hard limit count incremented by one

If within (soft limit amount) of hard limit, bring up hard limit message

- Nodes left before hard limit reached

If hard limit reached

- Message for hard limit reached
- Trigger recaptcha

Error Conditions:

1. Already existing node with inputted title
 - a. Error message
 - b. Node not created
 - c. User can change title
 - i. If title changed, check again, if good then node is created
 - d. If User exits function, data is deleted
2. Loss of internet connection
 - a. Error message
 - b. Maintain content data until User leaves function or reconnects
 - c. If User leaves function before reconnect, node is deleted
3. User hard limit reached
 - a. Error message when User clicks create button
 - b. User prevented from creating node

Non-Functional Requirements:

User can begin filling out input fields within 5 seconds

User tree data updated within 5 seconds

User tree page visually updated within 5 seconds

Use Case 2: Copying a node

Actors: User

Pre-Conditions:

Stable internet connection

User is logged in

User is on another User's tree page

Functions:

1. User left clicks a node
2. User left clicks copy button

Variations:

1. User holds shift and left clicks multiple connected nodes
 - a. User left clicks copy button
2. User left clicks select descendants button
 - a. User left clicks copy button
3. User has nodes selected to be copied
 - a. User left clicks a selected node to remove it from selection
 - b. If removed node has descendants, those descendants are also deselected
 - c. User left clicks copy button

Post Conditions:

User copy data updated with all selected nodes

Error Conditions:

1. Loss of internet connection
 - a. Error message
 - b. User copy data does not store the selected nodes
 - c. User needs to relick copy when reconnected to internet in order to store data

Non-Functional Requirements:

Copy data will be updated within 5 seconds

Use Case 3: Pasting a Node

Actors: User

Pre-Conditions:

Stable internet connection

User is logged in

User is on own tree page

Number of nodes created by the User has not reached the hard limit

Functions:

1. User right clicks open space
2. User left clicks paste button

Variations:

1. User left clicks a node
 - a. User left clicks paste button
 - b. Selected node is made parent of top level node in copy data
 - c.

Post Conditions:

User tree data updated with data of all nodes in copy data

- All copied nodes titles have original creators name appended
- All nodes in copy data placed onto User tree page

Copy data cleared

User tree data updated

User soft limit count incremented by number of nodes in copy data

If within 5 of soft limit, bring up soft limit message

- Nodes left before soft limit reached

If soft limit reached or exceeded

- Message for soft limit reached
- Trigger recaptcha
- Message for how many soft limits left, how many nodes left before hard limit
- If limit exceeded, number of exceeded carries over to next soft limit

User hard limit count incremented by number of pasted nodes

If within (soft limit amount) of hard limit, bring up hard limit message

- Nodes left before hard limit reached

If hard limit reached

- Message for hard limit reached
- Trigger recaptcha

Error Conditions:

1. Loss of internet connection
 - a. Error message
 - b. Paste prevented
 - c. Copy data is not deleted
 - d. User needs to relick paste once reconnected in order to paste nodes
2. Hard limit would be exceeded
 - a. Error message, notify User of how many nodes left before hard limit would be reached
 - b. Prevent paste
 - c. Clear copy data

Non-Functional Requirements:

User tree data updated within 5 seconds

User tree page visually updated within 5 seconds

Use Case 4: Setting node to private/public state

Actors: User

Pre-Conditions:

- Stable internet connection
- User is logged in
- User is on their own tree page
- User tree page has nodes

Functions:

1. User left clicks a node
2. User toggles private button
3. User left clicks save button

Variations:

1. Selected node has descendants
 - a. All descendents private/public state changed to match selected node's state
2. User holds shift and left clicks multiple connected nodes
 - a. Same as default

Post Conditions:

- User tree data updated with state change of selected node(s)

Error Conditions:

1. Loss of internet connection
 - a. Error message
 - b. Prevent User tree data update
 - c. User must re-click save button in order for tree data to be updated

Non-Functional Requirements:

- User tree data updated within 5 seconds
- User tree page visually updated within 5 seconds

Use Case 5: Changing contents of a node

Actors: User

Pre-Conditions:

- Stable internet connection
- User is logged in
- User is on their own tree page
- User tree page has nodes

Functions:

1. User left clicks on node
2. User left clicks edit button
3. User modifies input field contents
 - a. Title
 - b. description
4. User left clicks save button

Variations:

Post Conditions:

User tree data updated with changed content of selected node

Error Conditions:

1. Loss of internet connection
 - a. Error message
 - b. Prevent User tree data update
 - c. User must re-click save button in order for tree data to be updated

Non-Functional Requirements:

User tree data updated within 5 seconds

User tree page visually updated within 5 seconds

Use Case 6: Changing tag of a node

Actors: User

Pre-Conditions:

Stable internet connection

User is logged in

User is on their own tree page

User tree page has nodes

Functions:

1. User left clicks on node
2. User adds tags
3. User removes tags
4. User left clicks save button

Variations:

1. User holds shift and left clicks multiple connected nodes
 - a. Tags shown are based on last selected node
 - b. User adds tags
 - i. If a node already has the given tag, no changes to that nodes tags are made
 - c. User removes tags
 - i. If a node does not have the given tag, no changes to that nodes tags are made
 - d. User left clicks save button

Post Conditions:

User tree data updated with changed tags of selected node(s)

Error Conditions:

1. Loss of internet connection
 - a. Error message
 - b. Prevent User tree data update
 - c. User must re-click save button in order for tree data to be updated

Non-Functional Requirements:

User tree data updated within 5 seconds

User tree page visually updated within 5 seconds

Use Case 7: Changing parent of a node

Actors: User

Pre-Conditions:

- Stable internet connection
- User is logged in
- User is on own tree page

Functions:

1. User left clicks on node
2. User left clicks on change parent button
3. User selects new parent from list of existing nodes on tree page
4. User left clicks save button

Variations:

1. User selects "none" for parent
 - a. Selected node becomes the root node of a new tree

Post Conditions:

- User tree data updated with change to selected nodes parent
- If selected node has descendants, they move visually with the selected node

Error Conditions:

1. Loss of internet connection
 - a. Error message
 - b. Prevent parent change
 - c. User must re-click save button after reconnection to change parent

Non-Functional Requirements:

- User tree data updated within 5 seconds
- User tree page visually updated within 5 seconds

Use Case 8: Deleting a node

Actors: User

Pre-Conditions:

- Stable internet connection
- User is logged in
- User is on their own tree page
- User tree page has nodes

Functions:

1. User left clicks on a node
2. User left clicks delete button
3. User left clicks confirm button

Variations:

1. User holds shift and left clicks multiple connected nodes
 - a. User clicks delete button
 - b. User clicks confirm button
 - i. All selected nodes will be deleted
2. User left clicks select descendants button
 - a. User left clicks a node
 - b. User clicks delete button

- c. User clicks confirm button
 - i. All selected nodes will be deleted

Post Conditions:

1. Selected nodes are removed from user tree data
 - a. If selected nodes had descendants that were not selected for deletion, parent(s) of highest level descendant(s) changes to parent of selected highest level selected node
 - i. If it had a parent, User tree data updated with changes to highest level descendant(s) parent(s) (has a new parent)
 - ii. If it did not have a parent, User tree data updated with changes to highest level descendant(s) parent(s) (no parent)
2. If the User's hard limit count is greater than
 - a. Subtract one from the hard limit count for each deleted node
 - i. Stop of hard limit count reaches zero
 - b. Reduce soft limit count according to amount that the hard limit count is reduced
 - i. If soft limit count would reach zero, decrement soft limit number and repeat process with remaining hard limit count reduction

Error Conditions:

1. Loss of internet connection
 - a. Error message
 - b. Prevent User tree data update
 - c. User must re-click save button in order for tree data to be updated

Non-Functional Requirements:

- User tree data updated within 5 seconds
- User tree page visually updated within 5 seconds

Use Case 9: Searching for a topic

Actors: User

Pre-Conditions:

- Stable internet connection
- User is logged in

Functions:

1. User clicks on search input field
2. User types in phrase/topic to search for
3. User clicks enter key or search button

Variations:

1. User clicks on search input field
 - a. User types in phrase/topic to search for
 - b. User clicks filter button
 - i. User selects filters
 - c. User clicks enter key or search button
2. User clicks on phrase/topic provided in recent search list
 - a. Input field auto fills with phrase/topic
3. User clicks on phrase/topic provided in other lists
 - a. Automatically searches

Post Conditions:

1. System returns a list of Users whose tree page contains a public node titled exactly with

- the searched phrase/topic (ignoring casing), filtered by selected filters
- 2. System search data updated
 - a. Number of times searched phrase/topic has been searched for is incremented by 1

Error Conditions:

- 1. No results for searched phrase/topic
 - a. No results found message
- 2. Loss of internet connection
 - a. Error message
 - b. Prevent search
 - c. User must re-click search button after reconnection in order to complete search

Non-Functional Requirements:

Return search results within 5 seconds

Use Case 10: Rating a node

Actors: User

Pre-Conditions:

- Stable internet connection
- User is logged in
- User is on another User's tree page
- Other User's tree page has public nodes

Functions:

- 1. User left clicks on node
- 2. User left clicks on rate button
- 3. User selects rating
- 4. User left clicks save button

Variations:

- 1. User hold shift and left clicks multiple connected nodes
 - a. User left clicks rate button
 - b. User selects rating
 - c. User clicks save button
- 2. User left clicks selected descendants button
 - a. User left clicks a node
 - b. User left clicks rate button
 - c. User selects rating
 - d. User clicks save button

Post Conditions:

Other User tree data updated with changes to selected nodes ratings

Error Conditions:

- 1. Loss of internet connection
 - a. Error message
 - b. Prevent rating change
 - c. User must re-click save button in order to complete rating change

Non-Functional Requirements:

Other User tree data updated within 5 seconds

Other User tree page visually updated within 5 seconds