Unit Testing Report

Product Name: NeatTabs
Team Name: Tab-O-Rama
Date: 11/23/2016

Unit Tests

For unit tests, we severely underestimated the amount of time and organization required for implementing unit tests in our extension. One of the biggest hurdles was having to unit test the extension as a whole. Because we didn't start our unit tests from the beginning of sprint 1, it was difficult to implement testing without restructuring a good amount of our code. We are using Jasmine, which is a behavior-driven development framework for testing JavaScript code.

Arthur:
For my modules, I mainly covered testing the basic save and restore functionality of our extension. The modules mainly pertained to restore_tabs(), clear_tabs(), save_tabs(), export_tabs(), and the visualization.js file.

- restore_tabs()
    - Function Description: Takes a session and reopens the URLs in a new window
    - Legal inputs: valid URLs
    - Illegal inputs: invalid URLs
    - Expected Functionality: Chrome will take the session object and attempt to reopen the urls in a new tab. The outcome depends largely on what gets stored (whether our app is storing valid URLs or if that URL is a valid address)
- clear_tabs()
    - Function Description: Clears any information stored in chrome.storage.local
    - Legal inputs: items/objects stored in chrome.storage.local
    - Illegal inputs: n/a
    - Expected Functionality: Should call the Chrome API (chrome.storage.clear) to remove all items stored in local storage
- save_tabs()
    - Function Description: Saves all tabs open in Chrome
    - Legal inputs: chrome.tabs.query() API response
    - Illegal inputs: invalid/mutated chrome.query responses or null
    - Expected Functionality: The function will call chrome.storage.local.set() on the objects return in chrome.tabs.query(). This will be storing the query object in local storage as our session object.

- export_tabs()
    - Function Description: Exports a .csv file of the saved session
    - Legal inputs: session object URLs
    - Illegal inputs: null
    - Expected Functionality: Function will call the session object from local storage and create an array of all the URLs in the object. Then it will create a download link variable appending all the URLs to the variable and call chrome.downloads() to download the file.
- Visualization.js
    - Function Description: Displays a pie chart of time spent on the user’s domains
    - Legal inputs: time/domains tracked by timer objects
    - Illegal inputs: missing times or domains
    - Expected Functionality: Uses the google.charts library to display the times and domains tracked by timer.js. It takes in the domain and corresponding time, sorts them from most time spent to least, and then displays it as a pie chart in the options page.

Sean:
The modules I worked with were primarily in the $popup.js$.

- create_current_table()
    - Functionality: Displaying the current tabs open, numbering them, and attaching a button from inclusion/exclusion.
    - Illegal Input: Null/invalid/none
    - Legal Input: An array of tab objects inputted from the google chrome tabs array.
- destroy_saved_table()
    - Functionality: Erases the current saved tabs table for clearing storage or displaying a new table of saved tabs.
    - Illegal Input: null
    - Legal Input: The saved_table.
- save_tabs()
    - Functionality: Stores the user's selected current tabs, checking which tabs the user has chosen for storage. Saved with the key “saved_tabs”
    - Illegal Input: null
    - Legal Input:Chrome’s array of tabs that are currently opened.
- clear_storage()
    - Functionality: Erases storage.
    - Illegal Input: null
    - Legal Input: The local storage.
- excludeCurrentTab()
    - Functionality: Flags a current table in the current tabs table to be exclude from a future save; also serves as an inclusion if clicked again.
    - Illegal Input: null
    - Legal Input: A button click on a current table in the current tabs table.

- removeSaveTab()
    - Functionality: When a user clicks a button associated with this tab, this tab is a removed from a session, and the array is updated.
    - Illegal Input: null
    - Legal Input: A button click on a saved tab in the saved tabs table.

Gerardo: Most of the work I did was in timer.js. For my part I performed Test object for *timer.* The primary goal was to establish and validate the input at each stage.

- timer() Equivalence Class
    - Function Description: Create an object the has the following values: domain, startTime, time, storeTime and idle.
    - Legal inputs: The initial time when a time was created. How much time the given domain has been open. The store time on local storage and the state of the current url.
    - Illegal inputs: negative time for the time values and invalid url for the domain.
    - Expected Functionality: The expected functionality of the time object is to hold the values of the current url running.

- Addtime() Equivalence Class
    - Function Description:Add the time the current focus domain has been active since the last event listener was fired.
    - Legal inputs: the legal input should be a positive value.
    - Illegal inputs: illegal input will be negative value for the time.
    - Expected Functionality: The expected functionality of the add is to correctly add the time and the store time for the current active domain before changing to another to another domain.
- setCurrent() Equivalence Class
    - Function Description: Set the current domain the was specified by the getCurrentFocus.
    - Legal inputs: The legal input for this function should be a valid domain.
    - Illegal inputs: Illegal input for this function will be an undefined domain.
    - Expected Functionality: The expected functionality for this function should be to set the current active domain return by the getCurrentFocus.
- updateCurrentFocusTab() Equivalence Class
    - Function Description: This function returns the current active domain.
    - Legal inputs: legal input for this function should be either null or the current domain.
    - Illegal inputs: Illegal input for this function is an undefine domain rather than null or a valid domain .
    - Expected Functionality: The expected functionality of this function is to return a valid url or a domain.

- getFromStorage() Equivalence Class
    - Function Description: Call the google API for retrieving the store domain and time.
    - Legal inputs: legal input for this function should be the correct time return that is stored in the local storage.
    - Illegal inputs: Illegal input for this function will be returning a negative value from storage.
    - Expected Functionality: The expected functionality of this function is to correctly retrieve the current domain time.
- saveToStorage() Equivalence Class
    - Function Description: This function should save the time return by addtime.
    - Legal inputs: Legal input for this function should be a positive value.
    - Illegal inputs: Illegal input for this function is when the time save is negative or 0.
    - Expected Functionality: The expected functionality of this function is to save the time return by add to the local storage.

Richa: none