**Reddit.com Web Testing**

Carmen Condeluci, Kofi Osei

CS 1632 – DELIVERABLE 3: Web Testing with BDD

https://github.com/CCondeluci/CS1632-Deliverable3

**SUMMARY AND CHALLENGES**

For this deliverable, my partner and I have decided to test a subset of functionality on Reddit, as it is a service that is free, has easily create-able and disposable user accounts, and is something we are both relatively familiar with. Although Reddit does make heavy use of dynamic content, the presence of physical page links makes it easy for the Selenium WebDriver to be able to simply target objects on any given page. In addition to this, Reddit also has a definitive split in features that are available to users and moderators of subreddits, giving us a wide variety of features to choose from creating user stories and scenarios for each side.

However, this project was not without its challenges. Although creating user stories and scenarios was relatively simple, targeting and locating some specific objects on pages proved to be very difficult. The first of these were thread comments and replies. Their IDs and names are dynamically generated and multiple hidden textareas of the same name appear on every thread page, even on permalinked comment pages! To get around this, we were able to exploit the “edit” and “reply” button’s auto-focus functionality in order to dynamically target the textarea in question with dom=document.activeElement.

The second major difficulty was handling CAPTCHA codes required for users and moderators to submit text and link posts. In the IDE, we were using getEval calls to supply input to Selenium manually, but these do not function correctly within JUnit, neither as WebDriver nor WebDriver Backed exports. For JUnit, we found it best to simply use a JOptionPane InputDialog box as it freezes execution until the tester manually supplies the correct CAPTCHA code. Reddit’s CAPTCHA codes are particularly terrible, so it causes some difficulty when the tester inputs the wrong code as that test will immediately fail. However, devising a solution for this seemed to be far more trouble than it was worth, as running the test only takes a few seconds.

The last major challenge was creating a method of running the JUnit tests in one persistent Selenium WebDriver session. By default, the WebDriver runs each test in a separate, non-persistent web browser instance, closing each time a test is finished. For a site that is reliant on logins like Reddit and the types of tests we were performing, we found it best to use @BeforeClass and @AfterClass in order to set up a user or moderator and then close the WebDriver only once all tests for a given feature file were completed. Furthermore, as to ensure that the test suites are not dependent on one another or content that may or may not exist in the subreddit upon starting, it was necessary to include strict ordering to the JUnit tests via @FixMethodOrder. We are aware this is bad form, as tests are meant to be completely independent, but we found no other feasible way to test a site as complex as Reddit efficiently. Efforts have been made within the tests themselves when possible to make them as completely independent they can be (deleting posts/comments when finished, etc…).

The code for these JUnit tests can be found here:

<https://github.com/CCondeluci/CS1632-Deliverable3>

The subreddit we used for this test is: <https://www.reddit.com/r/cs1632seleniumtestbed/>

Accounts that were used for these tests were:

Username: cs1632\_carmenkofi Password: laboon

Username: cs1632\_carmenkofimod Password: laboon

**USER STORIES**

* 1. As a user
  2. I want to be able to log in to my account
  3. So I can view my post history
  4. As a user
  5. I want to be able to post links to images
  6. So other users can see and comment on my images
  7. As a user
  8. I want to be able manage my comments
  9. So I can express my opinions on posts freely
  10. As a moderator
  11. I want to be able to manage who can post in or read my subreddit
  12. So I can control which users can contribute to my subreddit
  13. As a moderator
  14. I want to be able to manage contributor posts
  15. So I can have full control over what posts appear

**FEATURE FILES**

1. Feature: Logging In
   1. Scenario: Successful Log In
      1. Given a correct username and password
      2. When the user tries to log in with these credentials
      3. Then the user is logged in and can view post history
   2. Scenario: Invalid Username
      1. Given an invalid username and any password
      2. When the user tries to log in with these credentials
      3. Then the user is alerted they have a wrong password
   3. Scenario: Invalid Password
      1. Given a valid username and an invalid password
      2. When the user tries to log in with these credentials
      3. Then the user is alerted they have a wrong password
   4. Scenario: No User Account
      1. Given the user has not created an account
      2. When the user tries to log in
      3. Then the user has the option to create an account
2. Feature: Posting Image Links
   1. Scenario: Successful Image Post
      1. Given a user has a link to an image
         1. And has typed a title
         2. And has correctly answered the CAPTCHA
      2. When the user submits the image link
      3. Then the user is directed to their posted image link
   2. Scenario: Invalid Post Title
      1. Given a user has a link to an image
         1. And has typed an invalid title
         2. And has correctly answered the CAPTCHA
      2. When the user submits the image link
      3. Then the user is alerted to fix the “title” field
   3. Scenario: Invalid CAPTCHA
      1. Given a user has a link to an image
         1. And has typed a title
         2. And has INCORRECTLY answered the CAPTCHA
      2. When the user submits the image link
      3. Then the user is alerted to attempt the CAPTCHA test again
   4. Scenario: Duplicate Link
      1. Given a user has a link to an image
         1. And has typed a title
         2. And has correctly answered the CAPTCHA
         3. And the link is already present in the subreddit selected
      2. When the user submits the image link
      3. Then the user is brought to the page of the existing link, alerting them it has already been submitted
   5. Scenario: Suggest Title
      1. Given a user has a link to an image
         1. And has NOT typed a title
      2. When the user clicks the “suggest title” button
      3. Then the post is given a title based on the link
3. Feature: Comment Management
   1. Scenario: Post Comment
      1. Given a user is on a post’s page
         1. And the user has typed any characters into the comment box
      2. When the user clicks the “save” button
      3. Then their comment is added to the post’s thread
   2. Scenario: Delete Comment
      1. Given a user is on a post’s page
         1. And the user has already commented on the post
      2. When the user clicks the “delete” button
      3. Then their comment is removed is removed from the post’s thread
   3. Scenario: Edit Comment
      1. Given a user is on a post’s page
         1. And the user has already commented on the post
      2. When the user clicks the “edit” button
      3. Then their comment is re-opened for editing
         1. And is saved upon clicking the “save” button
   4. Scenario: Reply Comment
      1. Given a user is on a post’s page
         1. And there exists comments on the post
      2. When a user clicks the “reply” button
      3. Then a dialogue appears for them to reply to a comment
         1. And is saved upon clicking the “save” button
   5. Scenario: Save Comment
      1. Given a user is on a post’s page
         1. And there exists comments on the post
      2. When a user clicks the “save” button
      3. Then the comment is added to their “saved” list
4. Feature: Manage Subreddit Contributors
   1. Scenario: Add Contributor
      1. Given a moderator is on the “contributors” page
      2. When a moderator adds a username to the approved submitters list
      3. Then that user is now an approved contributor
   2. Scenario: Remove Contributor
      1. Given a moderator is on the “contributors” page
         1. And contributors exist for the subreddit
      2. When a moderator clicks remove for any username
      3. Then that user is no longer an approved contributor
   3. Scenario: Message Contributor
      1. Given a moderator is on the “contributors” page
         1. And contributors exist for the subreddit
      2. When a moderator clicks “send message” for any username
      3. Then the moderator is directed to a page where they can send a private message to that user
   4. Scenario: Search Contributors
      1. Given a moderator is on the “contributors” page
         1. And contributors exist for the subreddit
      2. When a moderator searches for a username in the “jump to” box
      3. Then the moderator will be given a list of approved contributors that match their search
5. Feature: Manage Posts
   1. Scenario: Sticky Post
      1. Given a moderator is on a post’s page
      2. When a moderator clicks the “sticky this post” button
      3. Then the post will appear at the top of the subreddit automatically.
   2. Scenario: Unsticky Post
      1. Given a moderator is on a post’s page
         1. And that post is currently stickied
      2. When a moderator clicks the “unsticky this post” button
      3. Then the post will no longer appear at the top of the subreddit automatically.
   3. Scenario: Remove Post
      1. Given a moderator is on a post’s page
      2. When a moderator clicks the “remove” button
      3. Then the post will be removed from the subreddit
   4. Scenario: Approve Post
      1. Given a moderator is on a post’s page
      2. When a moderator clicks the “approve” button
      3. Then the post will be set to “approved” status
   5. Scenario: Approve Removed Post
      1. Given a moderator is on a post’s page
         1. And that post was previously removed
      2. When a moderator clicks the “approve” button
      3. Then the post will be made available again
         1. And the post will be set to “approved” status

**TEST SCREENSHOTS**









