**HSDeckAssistant Web Testing**

Carmen Condeluci, Andrew Harper

CS 1632 – DELIVERABLE 6

https://github.com/graphitezeppelin/cs1632-Deliv6

**SUMMARY AND CHALLENGES**

For this deliverable, we have decided to test a copy of Carmen Condeluci’s 1520 Web Applications project from this semester, HSDeckAssistant. This application is a deck creation, sharing, and collaboration platform for use alongside the popular online collectible card game Hearthstone, created by Blizzard. We have performed a set of tests through use of the BDD model as well as some generalized penetration testing. Since the site is one that was created within our group, we are able to freely do penetration tests.

**USER STORIES**

* 1. As a user
  2. I want to be able to log in to my account
  3. So I can view my deck lists
  4. As a user
  5. I want to be able create deck lists
  6. So I can share my deck lists with other users
  7. As a user
  8. I want to be able to filter the card list
  9. So I can more easily create deck lists with only cards I want
  10. As a user
  11. I want to be able to comment on and like deck lists
  12. So I can leave feedback for other users
  13. As a user
  14. I want to be able to update and revise my deck lists
  15. So I can keep my deck lists relevant
  16. As a user
  17. I want to be able to import and export deck lists
  18. So I can use deck lists with other tools easily

**FEATURE FILES**

1. Feature: Logging In
   1. Scenario: Successful Log In
      1. Given a correct email and password
      2. When the user tries to log in with these credentials
      3. Then the user is logged in and can view deck lists
   2. Scenario: Invalid Email
      1. Given an invalid email
      2. When the user tries to log in with these credentials
      3. Then the user is alerted their email is invalid
   3. Scenario: Invalid Password
      1. Given a valid email 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: Creating Deck Lists
   1. Scenario: Successful Creation
      1. Given a user creates a deck list
         1. And has added 30 cards
         2. And has added a title
      2. When the user saves the deck list
      3. Then the user is directed to their deck lists
   2. Scenario: Deck List Too Small
      1. Given a user creates a deck list
         1. And has NOT added 30 cards
         2. And has added a title
      2. When the user saves the deck list
      3. Then the user is alerted to add 30 cards
   3. Scenario: No Deck List Title
      1. Given a user creates a deck list
         1. And has added 30 crads
         2. And has NOT added a title
      2. When the user saves the deck list
      3. Then the user is alerted to add a deck list title
   4. Scenario: Add Card
      1. Given a user is creating a deck list
      2. When the user clicks on any card image
      3. Then the card is added to the user’s deck list
   5. Scenario: Remove Card
      1. Given a user is creating a deck list
         1. And the user has added at least one card to their deck list
      2. When the user clicks on the card in the deck list
      3. Then the card is removed from the user’s deck list
3. Feature: Posting Deck List Feedback
   1. Scenario: Comment on a Deck List
      1. Given a user is on a deck list page
      2. When a user types a comment into the box and clicks “Submit”
      3. Then a comment is posted by them on that deck list
   2. Scenario: Like a Deck List
      1. Given a user is on a deck list page
      2. When a moderator clicks “Click to Like”
      3. Then the user’s like is registered on that deck list
   3. Scenario: Un-Like a Deck List
      1. Given a user is on a deck list page
         1. And that user has already liked this deck list
      2. When a moderator clicks “Click to Unlike”
      3. Then the user’s like is un-registered from the deck list
4. Feature: Updating Decklists
   1. Scenario: Open Correct Editing Page
      1. Given a user in on their profile page
         1. And that user has already created a deck list
      2. When a user clicks the “Edit” link
      3. Then they are taken to the editing page with all information filled in
   2. Scenario: Edit Deck List Title
      1. Given a user is on a deck list’s editing page
      2. When a user edits the title of their deck list and saves it
      3. Then the title of the deck list reflects their edit
   3. Scenario: Edit Deck Contents
      1. Given a user is on a deck list’s editing page
      2. When a user edits the contents of the deck list and saves it
      3. Then the contents of the deck list reflect their edit
   4. Scenario: Edit Deck Description
      1. Given a user is on a deck list’s editing page
      2. When a user edits the description of a deck list and saves it
      3. Then the description of the deck reflect their edit
5. Feature: Filtering the Card List
   1. Scenario: Filter by Name
      1. Given a user is creating a deck list
      2. When the user types a card name into the search box
      3. Then that card is shown alone in the card list
   2. Scenario: Filter by Type
      1. Given a user is creating a deck list
      2. When the user types a card type into the search box
      3. Then only cards of that type are shown
   3. Scenario: Filter by Tribe
      1. Given a user is creating a deck list
      2. When the user types a card tribe into the search box
      3. Then only cards of that tribe are shown
   4. Scenario: Filter by Mechanics
      1. Given a user is creating a deck list
      2. When a user types a card mechanic into the search box
      3. Then only cards having that mechanic are shown
   5. Scenario: Filter by Class/Neutral
      1. Given a user is creating a deck list
      2. When a user clicks the class/neutral card button
      3. Then only class or neutral cards are shown
6. Feature: Importing and Exporting Decklists
   1. Scenario: Import a Valid Deck List
      1. Given a user is creating a deck list
         1. And the user clicks the “Import” button
      2. When a user enters a valid set of cards and clicks “Submit”
      3. Then those cards are imported into the deck list
   2. Scenario: Import Oversized Deck List
      1. Given a user is creating a deck list
         1. And the user clicks the “Import” button
      2. When a user enters a valid set of more than 30 cards and clicks “Submit”
      3. Then only the first 30 cards are imported into the deck list
         1. And the user is notified which cards were not added
   3. Scenario: Import Invalid Deck List
      1. Given a user is creating a deck list
         1. And the user clicks the “Import” button
      2. When a user enters an invalid set of cards and clicks “Submit”
      3. Then only valid cards will be added to the deck list
   4. Scenario: Export Deck List
      1. Given a user is creating or viewing a deck list
      2. When a user clicks the “Export” button
      3. Then the deck list is shown in Cockatrice format.

**TEST SCREENSHOTS**







