Testing Documentation

Cheng Zhou, Liam McChesney, Daniel Medvedev, Diego Marrero, Louis Marfone, Matthew Kachensky

Login:

- General Description
    - The login page will serve as the access point for users to enter their personal page. The page will be navigable from the website's home page. Failure to provide correct account information will recommend the user to navigate to the registration page or reset their accounts password.
- User exists (verifying email with database)
    - Is the input a **username** or **email**
    - Password matches (verify if hashed password matches the database).
- User does not exist
    - Redirects to registration page
- Testing
    - The test data will be a combination of username/emails/passwords.
    - Verify that regex matches registration requirements
        - Valid username/password but with additional characters or whitespaces
        - Capitalization
        - Unique characters
    - Example Test Data:
        - Usernames - checks that it exists in the database
            - AllowedUser - verifies associated password and can log in.
            - NotAllowed - notifies username/password is incorrect, then redirects to registration page
        - Emails - exists in the database
            - firstlast@colorado.edu - verifies password and logs in
            - newuser@colorado.edu - recommends registration page, notifies that email/password is incorrect
        - Passwords
            - AcceptablePassword123! - If matches with username/email logs in
            - UnacceptablePassword123! - informs users that username/email and password are incorrect.
- The QA environment will have a different database than the dev/local environment.
- Results
    - Correct username/email and password
        - Should be able to access user specific home page (successfully log in)
    - If username/email does not exist in the database

        - Redirect to registration page
    - Username/email exists in database but password does not match/exist
        - Send an ejs error message asking for the correct password
        - On third incorrect attempt offer password reset link in error message
- Testers
    - Those not involved with the project to get a new set of eyes on the website and its features. Will provide information about user tendencies which will help improve website functionality.
        - Classmates
        - Friends

Search:

- Should be able to return list of tickets that match keywords in search
    - Clicking on ticket would show details about it
        - Price
        - Resell
        - Who's selling
        - etc.
- Filter works properly
    - Filter by type (resell, official)
    - Filter by type of event
    - Filter by price range, allows user to type in price range
        - Only accepts ints: "i.e. 20 - 2000"
        - Limits price range to something reasonable (min $1, max $10000)
- Search accepts special characters, whitespaces, numbers, etc.
    - "Event 4092 !!" -> valid, returns some list of tickets with matching characters
    - "this%search%still%works" -> valid, accepts special characters
    - "" -> return nothing
    - "Some prompt with a lot of characters" -> Should limit prompt to be ~2000 characters max, anything less is valid
- Search returns empty list when invalid or when tickets don't exist
- Tests should be random combinations of words, special characters, and whitespaces
    - Test whether the search request returns a result for the search or not
- QA environment has different data for testing, as well as dev, both with different configs
- Test results:
    - Search is valid: Returns some list n of tickets that has keywords that match the search
    - Filter is valid: Return only tickets that match the filter
        - Type: Only tickets of that type are shown
        - Event: Only tickets of that event are shown

        - Price: Only tickets of that price range are shown
    - Search AND filter is valid:
        - Both features work simultaneously
- Testers:
    - Dev environment: Everyone working on the project
    - QA/Prod: Students from CU, or people on CU who want to buy/resell tickets

Register:

General description: This page should allow users to input a valid email address, username, and password. It would proceed to check if the user has already been created, if so it should tell the User to think over another Username or Email. After it would allow them to register and then if everything works out it will send them to the login page stating that the user was created.

Accepted Testers: People in our class.

Testing environment: This would be in QA because we want to check our user database to make sure that user's are not repeatedly creating the same profile and email. After that we could create dummy data where a user tries to register with a new Username but same email and that should be denied.

1. Email
    a. Checks for a valid cu boulder email. Also will check if the email already exists.
        - "blue@gmail.com" - Invalid email, wrong domain
        - "ThisIsAnEmail" - Invalid email, no domain
        - "Bad email@colorado.edu" - Invalid email, includes a space character
        - "BadEmail@colorado" - Invalid email, no ".edu"
        - "Wrong.edu" - Invalid email, no "@colorado"
        - "<-*current email in database*->" - Invalid entry, should tell the user that the email is already attached to an account
        - "<-*new valid email*->" - Valid entry should be accepted

2. Password
    a. Passwords should check for a capital, minimum of 8 characters, special character, no spaces, and a number.

    -Test cases:
        - "Abcds1!" - Not enough characters
        - "124382!!-as" - No capital
        - "Skobuffs$" - No number
        - "Skobuffs3" - No special characters
        - "Sko buffs3$" - Contains a space

        - “Skobuffs3$” - Valid entry should be accepted
3. Username
    a. Client should be able to make a valid username:
        - This username should not accept spaces
        - This would also check if we already have this username in the database.
        - Should also reject any special characters
            - “RedSocksFan12” - Valid username
            - “Red SocksFan12” - Contains a space
            - “SkoBuffs!!!!” - Contains a special character
            - “<-*current username in database*->” - Invalid entry, should tell the user that the username already exists
4. Confirm Password
    a. Checks if this field is the same as the original password field
        - “<-*Correct duplicated password*->” -Valid entry
        - “<-*Incorrect duplicated password*->” -Invalid entry, should tell the user that the passwords do not match