**Depop Login Functionality – Test Plan and Test Completion Report**

**Introduction**

Depop website has user login functionality in its home page. This is very high-level test plan to test login functionality only.

As I don’t have any business or functional requirements for Login feature, I had to test the feature and noted the observations.

This document has not only high-level test plan items, it has sections usually found in test completion report.

**Testing Scope and Objectives**

Scope of testing is limited to login functionality of Depop website <https://www.depop.com/>.

Below are the key functions that are in scope for login functionality

* User should be able to login with valid login credentials
* User should not be able to login with invalid login details
* User should get appropriate error messages when attempted to login with invalid login details
* User should be warned about user name and password fields being left empty
* User should be able to toggle between ‘Show Password’ and ‘Hide Password’ options with the Password field

**Not in Scope**

All other links and functionality apart from login feature are out of scope for this technical testing task. For example, below are out of scope for this test technical task

* Forgot Password link
* Facebook button
* Sign up button
* Menu link or any other links or buttons in the page

**Test Levels**

Below are the various test levels that will be performed for the Depop login feature technical task.

**Unit Testing (White Box Testing)**

* Field level validations (field lengths, allowed and not allowed characters including special characters, blank spaces) can be tested for Username and Password fields.
* API testing to be done to verify the login system responses for various combination of input values.

**Integration Testing**

* As testing of login feature is the only one in scope, integration testing is not relevant for this exercise.

**System Testing (Black Box Testing)**

* When the User Interface is ready, features specified in the above scope section will be tested using the planned test cases.

**User Acceptance Testing**

* The functionality in scope is relatively small. However, a high level of test cases can be designed to be run by end user to provide confidence in the delivered functionality.

**Automation Testing**

Due to the limitations of the login attempts for login only 5 input validations are done for automation.

Feature file is pretty self-explanatory and all the scenarios that are tested are clearly explained.

Parameterising has been done in order avoid duplication of code.

Asserts can be found in Depop\_Login.java

**Automation Testing Report**

* Automation of the key features of this login application is completed (the automation work is uploaded in github).
* Below test methodologies are used in automation framework
  + BDD Agile methodologies, having feature files
  + Selenium Webdriver
  + Cucumber
  + Java
* Automation test principles
  + Each test has a concise verification (as atomic as possible)
  + README.md file has all the details including how to run the test.
  + CI/CD – Scheduling automation run every time a new application build is deployed will help to identify the issues early.
* Test Automation Report
  + Report can be found within the automation project. Instructions on how to open the report are given in project README.md.
  + Report will be updated with new results every time you run the tests.

**Cross browser/Cross platform Testing**

* Desktop
* Device Android
* Device iOS
* Mac
* Windows
* Different browsers including Chrome, Firefox, Internet Explorer and etc

**Regression Testing**

* As this is the first iteration of testing, regression is not in the scope here. If any changes are made to this application due to bug fixes or functional enhancements, a set of tests need to be executed to ensure core existing functionality is not affected due to the changes introduced.

**End to End Testing**

* Usually this type of testing is performed after completing system testing where multiple systems with different interfaces are integrated. In this case, if API testing is performed initially to test the backend for login in large extent, then front end testing will have covered majority of the rest of the usability perspective.

**Security Testing**

* Session management: Once user is on logged in page, it is never expiring the session. This is just an observation.
* We are able to generate the below error as part of security testing
  + 400 invalid request
* We could not generate the below errors as part of security testing. This can be done with cutting or delaying the connection to the server on site.
  + 408 request time-out
  + 500 server error
* As there is not any access management, security testing in access to the application perspective is not done.
* Other security testing methods like brute force attack and SQL injection are not carried out as they are not applicable here.

**Performance Testing – Load and Stress**

* Load testing should be carried out to observe the application response times when it is under the expected load (For this task, as per login limitations couldn’t perform much)
* Stress testing is performed to verify the application behaviour under increased load (beyond usual expected load) to observe response times, how application can crash and recover, how failover and disaster recovery can be done. (Again, due to limitations of login attempts limit is required to be increased in order to perform this).

**Test Cases**

Should be in a very detailed format (If possible, get a test management tool like rally, testrail, etc) – Due to the time constraints, writing at high level.

However, a “Good Test Case” will have below:

* Preconditions => including test data creation, accessing environments, any required information to be able to execute the test
* Summary of test => Brief outline where you know what test is all about
* Steps for action => Detailed step wise action plan to execute the test case
* Expected Outcome => What is being expected of the action
* Any test data into the test case that any tester can pick up the test after say 6 months
* Any wireframe attachments that may be used for better understanding at the time of test

Tests below has a summary and are laid out as

<test case/scenario> - <an expected outcome> (Defect: if found)

**Acceptance criteria**

Entry criteria

* Unit tests are executed and all passed.
* Login backend implementation is done.
* Login front end website page is designed.
* Environments are available to deploy the feature to be tested.

Exit criteria

* All planned test cases are executed and passed.
* No high or medium priority defects are outstanding.
* Defect triage is done and little low priority defects are allowed.
* Test completion report is produced and accepted by product owner.

**Test Scenarios;**

1. Logging with correct username and password values – user logs in. (Automated)
2. Logging with correct email and password values – user logs in. (Automated)
3. Logging with incorrect email and password values – error message “invalid username and password” shown for user. (Automated)
4. Logging with incorrect username and password values – error message “invalid username and password” shown for user. (Automated)
5. Click login with empty username – error message “username is required” is shown for user. (Automated)
6. Click login with empty password and correct username – error message “password is required” is shown for user. (Automated)
7. Login with valid user credentials, logout and log back in. User is logged out and successfully will log back in.
8. Login with valid user credentials. Close and open the browser. User should have been logged in.
9. Login with valid user credentials. Open another tab with Depop link. User should have been logged in.
10. Enter valid user credentials. To click log in, use tab key and enter key when focus goes on login button. User should be able to login.
11. Enter the inactive and disabled login credentials and login. User should not be able to login.
12. Enter the valid username and an old password to login. User should not be able to login.
13. Enter the valid username and copy paste the password and login. Depop user is able to login.
14. Verification of cookie management should be verified here too.

**Exploratory Testing**

This is a type of testing where test cases are not created in advance but testers check system on the fly.

1. After user is logged in refresh the page - logged in session still persists
2. After user is logged in click back button on browser – logged in session still persists
3. In the browser, login in incognito mode and ensure user session details are deleted once the browser is closed.
4. Verify whether user can paste username and password values in to the fields.
5. Check whether the login details can be stored in various browsers.
6. Enter just numbers and click Login – error message “Invalid username and password” and sometimes “unknown error has occurred” like in below screenshot
7. Enter just special characters and click Login – “unknown error has occurred” like in below screenshot
8. Login with valid user credentials. Verify the session. User is logged in until user logs out themselves. (not sure if this is what it should happen)

Many more test cases can be written to have a thorough testing done for login page.

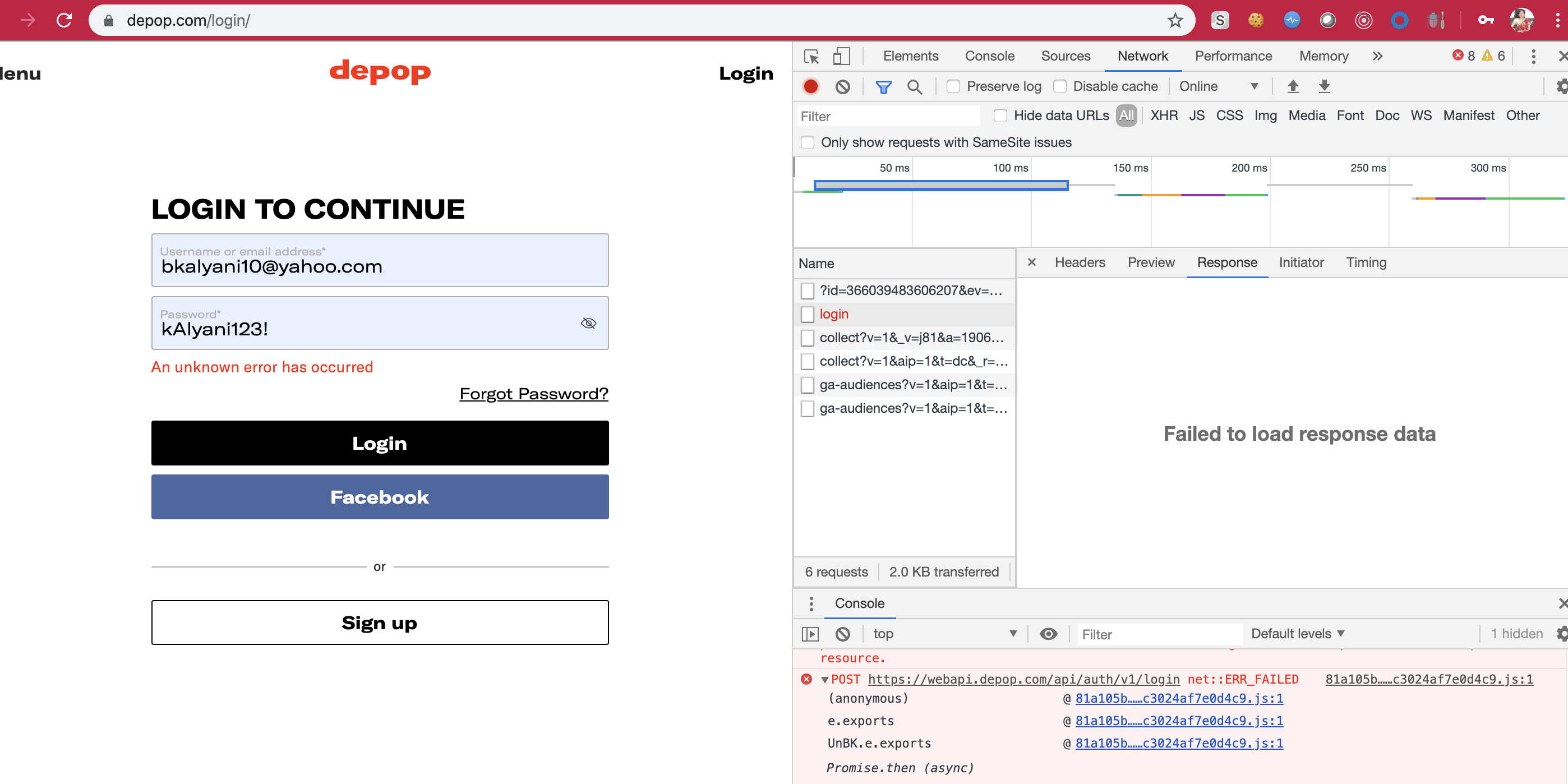
**Defects**

Ideally bugs to be raised in tools such as Jira, Bugzilla, etc where user stories exist.

However, a good defect should consist of the below:

* Summary of defect => A brief description as to understand what it is about
* Steps to Reproduce => A detailed step by step reproducible steps for a developer or any other team member to understand and reproduce
* Expected outcome => What was expected
* Actual Outcome => What is the outcome and why you think it is not correct
* Screenshots => Any screenshot that may help for more understanding of the defect
* Data => Any data that can be used to reproduce the defect

1. Enter correct username and wrong password and click login a few times – Error message appears “An unknown error has occurred” (Defect: User is left confused and the application continues throwing the same error even after entering valid login details) Screen shot below. There is a situation where user is thrown error message “unfortunately user is locked for xx hours”. It seems errors are not handled appropriately.



1. When I specify valid username or email address, enter a blank space in password field and click on Login button, the below error is displayed!

This Facebook account is not connected to Depop. You can do this through the Depop app. I couldn’t reproduce this defect again neither I took screenshot when it happened.

1. Separate Observation: While I was signing up, after populating Username and Password fields in 'Get Ready' page, I clicked on Next button and subsequently on 'Create Account' button in the next page. At this point, I am redirected to the 'Get Ready' page where I could see field level validations errors like the below. These should have been checked before user is allowed to the next page.

Help us protect you. Create a more secure password. Try using capital letters and numbers.

Ensure this value has at least 8 characters (it has 6).

**Test Completion Evaluation and Testing Performed**

High level tests are executed to cover all the above features listed in the test scope. All the observations during test execution are recorded.

As I am not working against any agreed requirements, based on common sense approach, certain observations are listed as defects in the below section.

To supplement the planned test cases, some more exploratory tests are executed and the results are below.

**Assumptions and Constraints**

* In the absence of exact requirement use cases, business requirements and functional requirements I assumed the generic application behaviour and written the high-level test cases.
* The expected results of the above test cases are based on generic assumption of how login functionality works.
* It’s assumed most of load, stress and non-functional testing is not in the scope of this assignment.
* The severity and priority of defects is not defined and it’s assumed that these defects will be triaged later.
* Due to the limitation where user can only login 10 time in an hour, I could not develop more than 10 automation scenarios that require login.
* Due to the limitation of login errors, throwing different errors, when continuous invalid login attempts are made, not been able to test scenarios as the login capability is pushed to hours to not been able to login up to 17 hours.
* Due to time constraints and lack of formal requirements, the test plan is not fully comprehensive.

**Residual Risks**

* As there are no load testing requirements, and limitations of the login attempts login implementation may need to be tweaked to be able to perform load testing.
* It seems login functionality is not stable. I could see many intermittent errors and the behaviour is not consistent when explorative testing is performed.
* Certain errors I witnessed could not be reproduced. A detailed investigation of application and sever logs need to be done to resolve these errors.

**Recommendations**

* All the defects need to be triaged with business before agreeing whether this application is fit for business purpose.
* The above **risks** should be considered before making any business decisions about suitability of this feature for the intended purpose.