UNIVERSITY OF CALGARY

ENSF 607

Advanced Software Development and Architecture
Testing Document for Stock Recommendation System

Objective:

To verify that the Stock Recommendation System functions correctly, we will develop a comprehensive testing plan, including unit tests, test data, and validation plans to ensure that the system performs as expected in terms of functionality and performance.

1. Validate Exchange Region - Exchange Not Recognized

- **Test Objective**: Verify that the system returns an appropriate error message when the exchange is not recognized.
- **Test Data:** Preferred exchange: UNKNOWN_EXCHANGE **,** Preferred region: North America
- **Test Expected Result:** The system should return the message: "Exchange 'UNKNOWN_EXCHANGE' is not recognized."
- **Actual Result:** The test correctly detected the unrecognized exchange and returned the expected error message.

2. Validate Exchange Region - Exchange Matches Preferred Region

- **Test Objective**: Ensure that no error is returned when the exchange matches the preferred region.
- **Test Data:**Preferred exchange: NASDAQ **,** Preferred region: North America
- **Test Expected Result:** The system should return None, indicating no mismatch between exchange and region.
- **Actual Result:** The test successfully confirmed that the exchange and region matched, returning no errors.

3. Validate Exchange Region - Exchange Mismatch with Preferred Region

- **Test Objective:** Verify that the system returns an error when the exchange does not match the preferred region.
- **Test Data:**Preferred exchange: NASDAQ, Preferred region: Europe
- **Test Expected Result:** The system should return the message: "The exchange 'NASDAQ' is not located in the preferred region 'Europe'."

- **Actual Result:** The system correctly identified the mismatch and returned the appropriate error message.

4. Get Recommendations from Gemini - Successful Response

- **Test Objective:** Ensure that the system can successfully fetch stock recommendations from the Gemini API when the response is valid.
- **Test Data:** Investment preference:
  Risk tolerance: 3 , Asset type: 1 (representing Stocks) , Preferred region: North America , Preferred exchange: NASDAQ
- **Test Expected Result:** The system should return a list of stock recommendations, including the symbols AAPL and GOOGL, along with their corresponding companies (Apple Inc. and Alphabet Inc.).
- **Actual Result:** The system returned the expected stock recommendations correctly, with AAPL and GOOGL and the associated companies.

5. Get Recommendations from Gemini - API Error Handling

- **Test Objective:** Ensure that the system raises an appropriate error when the Gemini API fails while fetching stock recommendations.
- **Test Data:** Investment preference as in the previous test case.
  Simulated error from Gemini API (mocked to throw an exception).
- **Test Expected Result:** The system should raise a ValueError indicating the failure due to the exception thrown by the Gemini API.
- **Actual Result**: The test correctly raised a ValueError when the mocked Gemini API failure occurred.

6. Get Recommendations from Gemini - Empty Response

- **Test Objective:** Ensure that the system raises an error if the Gemini API returns an empty response.
- **Test Data:** Investment preference as in the previous test case.Simulated empty response from Gemini (mocked to return an empty string).
- **Test Expected Result:** The system should raise a ValueError due to the empty response from Gemini.
- **Actual Result:** The test correctly raised a ValueError as expected due to the empty response.

7. Login with Valid Credentials

- **Test Objective:** Ensure that a user can log in with valid credentials.
- **Test Data:** Email: 'sanket@gmail.com', Password: sanket'.
- **Test Expected Result:** User is redirected to the investment preferences page with a success message: 'Login successful!'.
- **Actual Result:** Verified login and redirect to the investment preferences view.

8. Login with Invalid Credentials

- **Test Objective:** Ensure that an error message is displayed for invalid credentials.
- **Test Data**: Email: 'sanket@gmail.com',Password: 'wrongpassword'.

- **Test Expected Result:** The system should display an error message: 'Invalid username or password!' and redirect back to the login page.
- **Actual Result:** Verified error message and redirect.

9. Signup with Valid Data

- **Test Objective:** Ensure that a user can sign up with valid data.
- **Test Data:** Valid name, email, and password inputs.
- **Test Expected Result**: User is successfully registered, redirected to the login page, and sees a success message: 'User registered successfully!'.
- **Actual Result**: User created and redirected as expected.

10. Signup with Existing Email

- **Test Objective:** Ensure the system prevents duplicate registrations.
- **Test Data:** An email that already exists in the database.
- **Test Expected Result:** The system should display an error message: 'User already exists.'
- Actual Result: Verified error message for duplicate email.

11. Signup with Password Mismatch

- **Test Objective:** Ensure that the system handles password mismatches during signup.
- **Test Data:** Password and confirm password fields do not match.
- **Test Expected Result:** The system should display an error message: 'Passwords do not match.'
- **Actual Result:** Verified error message for mismatched passwords.

12. Signup with Invalid Email

● **Test Objective:** Ensure that the system handles invalid email formats during signup.
● **Test Data:** Email: 'invalid-email', Password: 'testpassword'.
● **Test Expected Result:** The system should display an error message: 'Email is invalid.'
● **Actual Result:** Verified error message for invalid email.

13. Signup with Empty Email Field

● **Test Objective**: Ensure that the system handles empty email fields during signup.
● **Test Data**: Email: '', Password: 'testpassword'.
● **Test Expected Result:** The system should display an error message: 'Email is invalid.'
● **Actual Result:** Verified error message for empty email fields.

14. Signup with Server Error

● **Test Objective:** Ensure that the system handles server errors gracefully during signup.
● **Test Data:** Valid name, email, and password inputs, with simulated server error.
● **Test Expected Result:** The system should display an error message: 'Something went wrong.'
● **Actual Result:** Verified error message for server-side failure.

15. Signup View Renders Template

● **Test Objective:** Ensure that the signup page renders correctly.
● **Test Data:** User accessing the signup page.
● **Test Expected Result:** The system should return a 200 OK response and render the signup.html template.
● **Actual Result:** Verified 200 OK response and correct template rendering.