test_plan_deliverable.md

Test Plan

Strategy

This purpose of this Test Plan is to ensure the quality of the App called **Holidays to Miami**.
To keep this document structured it will present 2 main categories. **Register** and **Login**.

Schedule, estimation and deliverables

This process will be performed in a full sprint (2 weeks), with a realistic estimation of half a spring (1 week).
The results of this plan, will be delivered through JIRA Xray Testing suite. The Test Plans will be detailed there, the Test Cases will be linked in the Test Plan, and finally the Test Execution Results will be displayed in the same structure (Visible in the Test Plan document).
The Testing Approaches will be 3 types of testing:

1. API Unit Tests
2. Manual UI testing
3. End to End Testing Automated Testing Solutions will be decided by the Team Assigned to this project.

Resources required

For this task, there will be 1 member from the Product Owners team that will provide the project requirements and will follow the process of this project. 2 QA members in charge of Testing and the Tech Lead, and a Senior Dev from the Dev team.
The journey will start with a meeting between all of the Project Members. There they will decide the Project's duration, the resources required (for example, more Dev members will be required to provide the App being tested, fixing bugs or defects, etc.), and the Testing Software solutions to employ. For example, they might want to use Cypress for End to End Testing.
EveryThere'll be a weekly standup, in it all members that are enrolled in this project will participate, the current progress and the planned roadmap be monitored and managed by stakeholders.

Register, Test Cases

0. Basic Requirements
    - i. It should
        - a. have Front End solution do the following
            - a. Each field that is required marked with an asterisk
            - b. A clear message near the Submit button explaining that the (*) fields are required
            - c. All fields with validation must have a toggle "validation requirements" small button underneath the field clearly explaining the validation rules that apply to the field
        - b. API/Back End should have rules for required and validation and proper response messages
1. Register with Empty Data
    - i. It should
        - a. have Front End solution do the following
            - a. avoid submitting the te form
            - b. report mandatory fields with
                - a. UI field highlighting
                - b. Error message explaining that the field is required
        - b. API should
            - a. return a status on the 400 range
            - b. return an error message saying all fields are empty
2. Register with all required fields filled in with space characters
3. It should
    - i. have Front End solution do the following
        - a. avoid submitting the te form
        - b. report mandatory fields with
            - a. UI field highlighting
            - b. Error message explaining that the field is required
    - ii. API should
        - a. return a status on the 400 range
        - b. return an error message saying all fields are empty
4. Register with all required (correct) fields except one
    - i. It should
        - a. Front End part
            - a. avoid submitting the form
            - b. report the mandatory field with
                - a. UI field highlighting

            b. Error message explaining that the field is required
    b. API should
        a. return a status on the 400 range
        b. return an error message naming the required field

5. Register with the required fields but with validation errors **REPEAT FOR EACH FIELD WITH VALIDATION**
    i. It should
        a. Front End part
            a. avoid submitting the form
            b. report the validation error fields with
                a. UI field highlighting
                b. Error message explaining the validation rules that apply and why they failed
        b. API should
            a. return a status on the 400 range
            b. return an error message naming the required, the validation rules it failed and why it failed
6. Register with all required (correct) fields except 3 of them with validation errors
    i. It should
        a. Front End part
            a. avoid submitting the form
            b. report the validation error fields with
                a. UI field highlighting
                b. Error message explaining the validation rules that apply and why they failed
        b. API should
            a. return a status on the 400 range
            b. return an error message naming the required, the validation rules it failed and why it failed
            c. ensure that all fields that failed are listed with their proper explanation of the error
7. Register with all required (correct) fields, but a few validation error in a non required field
    i. It should
        a. Front End part
            a. avoid submitting the form
            b. report the validation error field with
                a. UI field highlighting
                b. Error message explaining the validation rules that apply and why they failed

- b. API should
  - a. return a status on the 400 range
  - b. return an error message naming the required, the validation rules it failed and why it failed
  - c. ensure that all fields that failed are listed with their proper explanation of the error

8. Register with all required (correct) fields, non required fields empty
   - i. It should
     - a. Front End part
       - a. load the Login page
       - b. display a highlighted UI block element with the title "Register process complete!
       - c. the UI element should be clearly standing out from the login page visually
       - d. the UI element should provide an option to close it
     - b. API should
       - a. response with a 201 code for successfully created
9. Register with all required (correct) fields, non required fields too.
   - i. It should
     - a. Front End part
       - a. load the Login page
       - b. display a highlighted UI block element with the title "Register process complete!
       - c. the UI element should be clearly standing out from the login page visually
       - d. the UI element should provide an option to close it
     - b. API should
       - a. response with a 201 code for successfully created

Login, Test Cases

0. Basic Requirements
   - i. It should
     - a. have Front End solution do the following
       - a. provide clear instruction in the UI, like for example "enter credentials to login" or "enter user and password to login"
     - b. API/Back End should
       - a. avoid error responses detailing which specific field was not found or with errors
1. Sending empty form

   i. It should
      a. have Front End solution do the following
         a. avoid submitting the te form
         b. report mandatory fields with
            a. UI field highlighting
            b. Error message explaining that the field is required
      b. API should
         a. return a status on the 400 range
         b. return an error message saying credentials are empty

2. Sending form with only USERNAME
   i. It should
      a. have Front End solution do the following
         a. avoid submitting the te form
         b. report mandatory fields with
            a. UI field highlighting
            b. Error message explaining that the field is required
      b. API should
         a. return a status on the 400 range
         b. return an error message saying password is required
3. Sending from with only PASSWORD
   i. It should
      a. have Front End solution do the following
         a. avoid submitting the te form
         b. report mandatory fields with
            a. UI field highlighting
            b. Error message explaining that the field is required
      b. API should
         a. return a status on the 400 range
         b. return an error message saying username is required
4. Sending from with wrong **USERNAME**
   i. It should
      a. have Front End solution do the following
         a. credentials error with
            a. UI elments highlighting
            b. Error message simply stating "wrong credentials", to avoid specifying which field is right or wrong (basic security measure)
      b. API should
         a. return a status on the 400 range
         b. return an error message saying "wrong credentials"
5. Sending from with wrong **PASSWORD**

- i. It should
  - a. have Front End solution do the following
    - a. credentials error with
      - a. UI elments highlighting
      - b. Error message simply stating "wrong credentials", to avoid specifying which field is right or wrong (basic security measure)
  - b. API should
    - a. return a status on the 400 range
    - b. return an error message saying "wrong credentials"

6. Sending from with both correct credentials
   - i. It should
     - a. Front End part
       - a. load the Home page
       - b. display a highlighted UI block element with the title "Login successful!"
       - c. the UI element should be clearly standing out from the Home page visually
       - d. the UI element should provide an option to close it
         - a. consider if the UI element should fade away from visibility once a countdown has been reached to avoid residual elements in UI
     - b. API should
       - a. response within a 200 range code for successfully Logged In