Testing Specification

The Republic Project

Overview

This document outlines the testing strategies and methodologies to ensure the quality and reliability of The Republic project.

Contents

1 Introduction

Ensuring the quality and reliability of The Republic is paramount. This document details the testing strategies and methodologies used to validate the system's functionality and performance.

2 Testing Objectives

2.1 Performance Testing

- **Volume Handling:** Ensure the system can manage a large number of textual posts efficiently.
- **Media Uploads:** Assess the system's performance when users upload images to their posts.
- **Commenting:** Evaluate the system's ability to handle multiple users commenting on posts simultaneously.
- **Reactions:** Test the system's responsiveness when processing reactions to posts.

2.2 Reliability Testing

- **Account Creation:** Ensure users can create accounts reliably under various conditions.
- **Login:** Verify the reliability of the login process.

2.3 Scalability Testing

- **Filtered Posts:** Assess the system's ability to handle posts filtered by various criteria.
- **Role-Based Views:** Evaluate scalability when different user roles access the system.
- **Filtered Post Display:** Test the system's performance when displaying posts with multiple filters applied.

2.4 Security Testing

- **Account Updates:** Ensure the security of users updating their account details.
- **Role Selection:** Test the security of the role selection process during account creation.
- **Profile Access:** Verify that only authorized government officials can access user profiles.

2.5 Maintainability Testing

- **Profile Management:** Validate the ease of managing profile display names.
- **Anonymous Posting:** Assess the simplicity of posting anonymously and changing anonymity settings.
- **Role Changes:** Test the flexibility of changing user roles within profiles.

2.6 Usability Testing

- **Post Viewing:** Evaluate how easily users can view their posts on their profiles.
- **Issue Categories:** Test the user experience when selecting issue categories for posts.
- **Location Selection:** Assess the usability of selecting post locations.
- **Data Analytics:** Evaluate the clarity and usability of standard and filtered data analytics visualizations.

- **Report Creation:** Assess the ease of creating reports based on dates and locations.
- **Report Clarity:** Evaluate the clarity of reports with statistical analytics visualizations.

3 Test Plan

3.1 Overview

The test plan includes:

- Performance Testing
- Reliability Testing
- Scalability Testing
- Security Testing
- Maintainability Testing
- Usability Testing

3.2 Objectives

The tests will ensure that the final product functions as expected. Specifically:

- Frontend components display correctly.
- Backend and API endpoints return valid data for correct requests.
- Clear error messages are shown for incorrect requests.
- Proper integration between frontend and backend systems.

4 Test Cases

4.1 Performance Testing

4.1.1 Test 1: High Volume of Textual Posts

- **Objective:** Validate the system's handling of a large number of textual posts.
- **Procedure:** Simulate many users creating posts simultaneously.
- **Expected Outcome:** The system should remain fast and responsive without significant delays.

4.1.2 Test 2: Media Uploads

- **Objective:** Assess performance with concurrent media uploads.
- **Procedure:** Upload many images to posts at the same time.
- **Expected Outcome:** The system should handle uploads efficiently without slowing down.

4.1.3 Test 3: Concurrent Commenting

- **Objective:** Evaluate performance with multiple users commenting on the same post.
- **Procedure:** Simulate numerous users commenting simultaneously.
- **Expected Outcome:** The system should handle concurrent comments without delays or errors.

4.1.4 Test 4: Reactions to Posts

- **Objective:** Test performance when processing a high volume of reactions.
- **Procedure:** Generate many reactions (likes, dislikes) on posts.
- **Expected Outcome:** The system should remain stable and responsive.

4.2 Reliability Testing

4.2.1 Test 1: Account Creation

- **Objective:** Ensure reliable account creation.
- **Procedure:** Create accounts with various valid and invalid information under different network conditions.
- **Expected Outcome:** Accounts should be created consistently without errors.

4.2.2 Test 2: Login Reliability

- **Objective:** Verify reliable user logins.
- **Procedure:** Attempt logins with correct and incorrect credentials under normal and high load conditions.
- **Expected Outcome:** Users should log in reliably, and the system should handle logins well under heavy load.

4.3 Scalability Testing

4.3.1 Test 1: Filtered Posts

- **Objective:** Assess handling of filtered posts.
- **Procedure:** Apply various filters to posts and evaluate response time.
- **Expected Outcome:** The system should scale effectively for filtering without performance degradation.

4.3.2 Test 2: Role-Based Views

- **Objective:** Evaluate scalability with different user roles.
- **Procedure:** Simulate concurrent access by different user roles.
- **Expected Outcome:** The system should serve role-based views efficiently without performance issues.

4.3.3 Test 3: Filtered Post Display

- **Objective:** Test displaying posts with multiple filters.
- **Procedure:** Apply multiple filters simultaneously and assess response time.
- **Expected Outcome:** The system should handle concurrent filter requests without significant latency.

4.4 Security Testing

4.4.1 Test 1: Secure Account Updates

- **Objective:** Validate secure updates to account details.
- **Procedure:** Attempt updates with authorized and unauthorized access.
- **Expected Outcome:** Only authorized users should update account details.

4.4.2 Test 2: Secure Role Selection

- **Objective:** Test role selection security during account creation.
- **Procedure:** Attempt to manipulate roles during creation.
- **Expected Outcome:** Roles should be securely assigned and protected from unauthorized manipulation.

4.4.3 Test 3: Secure Profile Access

- **Objective:** Verify secure access to user profiles by government officials.
- **Procedure:** Attempt to access profiles as both authorized and unauthorized users.
- **Expected Outcome:** Only authorized officials should access user profiles.

4.5 Maintainability Testing

4.5.1 Test 1: Profile Display Names

- **Objective:** Validate ease of managing profile display names.
- **Procedure:** Attempt to change display names under various conditions.
- **Expected Outcome:** Users should easily update their display names.

4.5.2 Test 2: Anonymous Posting

- **Objective:** Assess ease of posting anonymously.
- **Procedure:** Test anonymous posting and changing anonymity settings.
- **Expected Outcome:** Users should easily post anonymously and change settings.

4.5.3 Test 3: Role Changes

- **Objective:** Test flexibility of changing profile roles.
- **Procedure:** Attempt to change roles and assess the process.
- **Expected Outcome:** Users should change roles easily without issues.

4.6 Usability Testing

4.6.1 Test 1: Viewing Own Posts

- **Objective:** Evaluate convenience of viewing own posts.
- **Procedure:** Navigate profiles and assess post visibility.
- **Expected Outcome:** Users should easily locate and view their posts.

4.6.2 Test 2: Issue Category Selection

- **Objective:** Test selecting issue categories when posting.
- **Procedure:** Attempt to select categories and assess usability.
- **Expected Outcome:** Users should intuitively choose categories.

4.6.3 Test 3: Post Location Selection

- **Objective:** Assess selecting post locations.
- **Procedure:** Test the process and evaluate usability.
- **Expected Outcome:** Users should select locations easily with clear indicators.

4.6.4 Test 4: Data Analytics Visualizations

- **Objective:** Evaluate viewing standard data analytics visualizations.
- **Procedure:** Access visualizations and assess user experience.
- **Expected Outcome:** Users should easily interpret and navigate visualizations.

4.6.5 Test 5: Filtered Data Visualizations

- **Objective:** Test viewing filtered data visualizations.
- **Procedure:** Apply filters and assess clarity of visualizations.
- **Expected Outcome:** Users should clearly view filtered visualizations.

4.6.6 Test 6: Report Creation

- **Objective:** Assess ease of creating reports based on dates and locations.
- **Procedure:** Generate reports with specific criteria.
- **Expected Outcome:** Users should easily create reports with desired criteria.

4.6.7 Test 7: Report Clarity

- **Objective:** Evaluate clarity of statistically shown reports with visualizations.
- **Procedure:** Access reports and assess clarity.
- **Expected Outcome:** Users should interpret reports easily with clear visualizations.

5 Automated Testing

5.1 GitHub Actions

We will use GitHub Actions for Continuous Integration (CI) and Continuous Deployment (CD):

- **Unit Tests:** Run Jest tests on pull requests to the `develop` branch.
- **Linting:** Check code compliance on pull requests to the `develop` branch.
- **End-to-End Tests:** Run Cypress tests on each pull request to the `main` branch.

5.2 Test Data

Various test data will be created to represent different user scenarios:

- Valid and invalid user credentials.
- Issue content with varying lengths and media attachments.
- Mock user data for interactions and relationships.

5.3 Methodologies

- **Unit Testing:** Test individual components in isolation.
- **Integration Testing:** Verify interactions between components and back-end API.
- **End-to-End Testing:** Simulate user interactions to test complete functionalities.

5.4 Tools

We will use the following tools for testing:

- **Frontend Testing:** Jest
- **Backend and API Testing:** Jest
- **End-to-End Testing:** Cypress