SOEN390 - Testing Plan For Sprint #2

Overview

Our testing strategy will encompass all levels of testing: unit tests, integration tests, and system tests, with an emphasis on automation to ensure efficient and reliable testing processes. The testing pyramid principle will guide our efforts, focusing on a larger number of unit tests, followed by integration tests, and finally, system tests to validate the end-to-end functionality of the application.

Tools and Technologies

- Unit Testing:
  - ReactJS: Jest coupled with React Testing Library for component and hook testing.
  - Django: pytest-django and Factory Boy for testing models, views, and Django REST framework serializers
- Integration Testing:
  - ReactJS + Django: Cypress for end-to-end testing scenarios that cover both frontend and backend integrations.
- System Testing:
  - Overall Application: Selenium WebDriver for comprehensive system testing, ensuring compatibility across different browsers and environments.

- Automation:
  - Continuous Integration (CI): GitHub Actions for automating tests upon every commit or merge request, ensuring all new changes are tested.

Unit Tests

- Objective: To validate the functionality of individual components (ReactJS) and modules (Django) in isolation from the rest of the system.
- Approach: Mock external dependencies using Jest mocks (for ReactJS) and pytest fixtures (for Django) to isolate the unit of work.

Integration Tests

- Objective: To verify the interactions between different parts of the application, such as React components calling Django REST APIs.
- Approach: Use Cypress to simulate user actions that trigger interactions between frontend and backend components, validating the integrated flow.

System Tests

- Objective: To evaluate the complete, integrated application's functionality in an environment that mimics production.
- Approach: Employ Selenium WebDriver to perform tests that cover user scenarios from start to finish, including interaction with all external integrations.

Acceptance Tests

- Objective: To confirm that the application meets the business requirements and is ready for deployment.
- Approach: Business Analysts and Product Owners will define acceptance criteria for user stories. These criteria will be translated into automated tests using Cypress, ensuring that critical user journeys are validated.

Metrics and Coverage Goals

- Aim for at least 80% code coverage across unit and integration tests.
- System test coverage should ensure all critical user paths are tested.
- Track bug density and fix rate post-testing to measure quality improvement.

Conclusion

This Testing Plan for Sprint #2 outlines a comprehensive approach to ensure that every aspect of the condo management app is thoroughly validated. The use of automated testing tools and techniques, aligned with the testing pyramid strategy, will help in maintaining high quality and reliability of the application as we progress through development sprints.