SOEN390 - Testing Plan for sprint #3

Testing Tools:

- ReactJS: Continue using Jest alongside the React Testing Library to handle component and hook testing. These tools are efficient for testing React components in a controlled environment.
- Django: pytest-django and Factory Boy remain the tools of choice for testing models, views, and Django REST framework serializers, ensuring that the backend logic and data handling are correctly implemented.
- Integration and System Testing: Cypress and Selenium WebDriver will be used for end-to-end testing scenarios that validate the integration between the frontend and backend, as well as the system's behaviour in an environment that closely mimics production.

Testing Approach and Coverage:

Unit Testing:

- Objective: Ensure that individual components and modules function correctly in isolation, with a focus on the new features developed during Sprint #3.
- Coverage Goals: Maintain or increase code coverage to above 80% for new and modified code, ensuring that new functionalities are thoroughly tested.

Integration Testing:

- Objective: Test the interaction between new and existing components, especially focusing on new APIs, data flow, and user interactions introduced in Sprint #3.
- Coverage Goals: Cover all new integration points and critical paths affected by recent changes, with special attention to reservation system enhancements and financial reporting features.

System Testing:

- Objective: Validate the application's overall functionality, user experience, and performance, ensuring that Sprint #3 additions integrate seamlessly with existing features.
- Coverage Goals: Execute comprehensive tests covering all user stories implemented in Sprint #3, including the reservation system's usability and financial report accuracy.

Acceptance Testing:

- Objective: Confirm that the application meets all updated business requirements and is ready for deployment, with a specific focus on user satisfaction and compliance with the updated product vision.
- Approach: Convert acceptance criteria defined by Business Analysts and Product Owners into automated tests, prioritizing critical user journeys and new functionalities introduced in Sprint #3.

Metrics and Reporting:

- Code Coverage: Aim for at least 80% coverage across unit and integration tests, with detailed reports on areas lacking coverage for targeted improvements.
- Bug Density and Fix Rate: Monitor and report on the number of defects per thousand lines of code and the rate at which they are resolved, aiming for continuous quality improvement.
- User Story Completion: Track the completion and acceptance testing success rate of user stories associated with Sprint #3, ensuring that all features deliver the expected value.

Additional Considerations:

- Continuous Integration (CI): Enhance CI pipelines to automatically run the expanded test suite against new commits or merge requests, incorporating static code analysis and security scanning as part of the build process.
- Test Data Management: Develop strategies for managing test data, especially for testing the financial system and reservation features, ensuring data integrity and relevance for realistic testing scenarios.
- Feedback Loops: Establish rapid feedback mechanisms for developers, testers, and stakeholders to address issues and improvements quickly, incorporating lessons learned into the testing strategy continuously.