

SOEN390 - Testing Plan

This testing plan dictates what will be done for sprint 3. We plan on having at least 80% code coverage.

Currently we are manual testing, which could be (partially) automated using continuous integration.

We plan on implementing a continuous integration pipeline that will check every merge against our test suite to ensure minimal mistakes and bugs.

We plan to write tests for all levels:

- Unit tests
- Integration tests
- System tests

The toolset of choice is Jest testing framework, which will be used for all our test suites

Unit Testing

In this phase, each unit is tested independently to ensure that it behaves as expected according to its specifications covering both typical and edge cases.

Reservation System:

- Test reservation creation.
- Test reservation update.
- Test reservation cancellation.
- Test reservation retrieval.
- Test reservation validation rules.

Requests Endpoints:

- Test request creation endpoint.
- Test request update endpoint.
- Test request cancellation endpoint.
- Test request retrieval endpoint.
- Test request validation rules.

Request Submitter Form:

- Test form submission with valid data.
- Test form submission with invalid data.
- Test form validation rules.

Database Web-Socket Connection for Notification:

- Test notification broadcasting on database changes.
- Test notification subscriptions.

Integration Testing

This section tests the integration between different components of the system, ensuring that they work together seamlessly to achieve the desired functionality. It covers scenarios where multiple components interact, such as the interaction between the reservation system and request endpoints or the integration between user roles management and other system components.

1. Test the integration between the reservation system and request endpoints.
2. Test the integration between request submitter form and requests endpoints.
3. Test the integration between request list view and database web-socket connection for notification.
4. Test the integration between user roles management and various system components.
5. Test the integration between admin role functionalities and user management components.
6. Test the integration between public user functionalities and request submission components.

System Testing

The system test plan focuses on testing the entire system as a whole, simulating end-to-end user interactions and scenarios. It verifies that all components work together harmoniously to fulfill the system's requirements and objectives. This includes testing the complete flow of various features from user interaction to backend processing and database operations.

1. Test end-to-end flow of the reservation system from creation to cancellation.
2. Test end-to-end flow of request submission, approval, and notification.
3. Test end-to-end flow of user roles management from creation to deletion.
4. Test end-to-end flow of admin role functionalities including access control and permissions.
5. Test end-to-end flow of public user functionalities such as request submission and viewing.

6. Test end-to-end flow of employee functionalities including request list viewing and management.