Testing Assignment

30 points

Shirel Snopik

Contents

[Introduction: 1](#_Toc108437046)

[Test Plan: 2](#_Toc108437047)

[User Stories: 2](#_Toc108437048)

[System (Feature) Tests: 3](#_Toc108437049)

[First User Story – Admin can create an order and save it: 3](#_Toc108437050)

[Second User Story – Admin can edit an order and save it: 3](#_Toc108437051)

[Unit tests: 4](#_Toc108437052)

[First User Story – Admin can create an order and save it: 4](#_Toc108437053)

[Second User Story – Admin can edit an order and save it: 4](#_Toc108437054)

[Completed tests: 5](#_Toc108437055)

[Automatic testing in GitHub: 5](#_Toc108437056)

[Testing in terminal: 6](#_Toc108437057)

[Evaluation: 7](#_Toc108437058)

[Mistakes Errors that can be detected 7](#_Toc108437059)

[Mistakes Errors that cannot be detected 7](#_Toc108437060)

[Conclusion 7](#_Toc108437061)

# Introduction:

This test plan will present the tests that I want to check on the orders creation and editing. An order is created and edited by an administrator, and later is sent to the production workers to be produced, for the order to be able to be produced its status needs to be “Production Pending” to get to this status there are some conditions.

The order contains a production machine, materials, and a pallet. These three fields are created using relationships to other tables, an order must have a pallet and a machine (even if the machine is called “none”) they cannot be left empty. An order can be created without any materials but then its status will be affected.

Orders statuses change automatically using event listeners:

“Admin Hold” - any criteria are missing (materials, machine is in “None “, and or start date for production is in null).

“Quality Check Pending” - If all the criteria above exists but the order dosent have a quality check attached to it.

“Production Pending” – the order has all the criteria mentioned above including the check and is ready for production.

An administrator can also cancel the order and then the status of that order will be “canceled” so any event listeners will not work anymore on this order until it will be reopened. And can also select and unselect an order, meaning when an order is selected the admin can view its full details and when its unselected, he can see all orders currently in the system in an overview.

These tests will only check the administrator side of the order handling and will focus on the procedure described above. The unit tests will check the event listeners and the smaller functions like selecting/unselecting and canceling an order. The feature tests check the storing and updating of orders if everything if correct and what happens if there are errors in the data filled.

The specification for the feature tests is based on the requirements and their acceptance tests as described in the next chapter, the fields were determined accordingly as well to the criteria and needs of the client. The unit tests check the detailed design, the business logic that is incorporated in the user stories to make them comfortable and usable by the users of the application.

# Test Plan:

## User Stories:

1. As administrative worker, I want to add a new order file to the system so that the production workers will know what pallets they should create in what amount.

**Happy Path:**

❖ GIVEN I’m on the orders’ view page

❖ WHEN I click the Add New button

❖ THEN a form page pops up where I can fill in the order details

❖ WHEN I click the Save button

❖ THEN the new order’s details view page will load

❖ WHEN I press the Enter Order View button

❖ THEN the new order’s details view page will load with the correct side menu to view all the orders extra details (drawings, checks, notes)

❖ WHEN I press the Back to Overview button

❖ THEN the orders index page will load with a new row for the new order

**Unhappy path:**

❖ GIVEN I’m on the orders’ view page

❖ WHEN I click the Add New button

❖ THEN a form page pops up where I can fill in the order details

❖ WHEN I click the Save button

❖ THEN Errors will show next to fields that were not filled properly or Required fields that were left empty

❖ WHEN I see the errors

❖ THEN I correct the filled in information in the necessary fields

❖ WHEN I click the save button

❖ THEN the new order’s detail’s view page will load

❖ WHEN I press the home page button

❖ THEN the orders view page will load with a new row for the new order

1. As administrative worker, I want to update an existing order’s details in the system so that the order files will be updated according to changes or additional requirements from the costumer.

**Happy Path:**

❖ GIVEN I’m on the orders’ view page

❖ WHEN I click the show button in the row of the desired order

❖ THEN the view page of the desired order’s details will appear on the screen

❖ WHEN I click the Edit button

❖ THEN a form page pops up where I can fill in and update the order details

❖ WHEN I click the Save button

❖ THEN the view page of the order’s details will appear on screen with the updated

details

**Unhappy path:**

❖ GIVEN I’m on the orders’ view page

❖ WHEN I click the show button in the row of the desired order

❖ THEN the view page of the desired order’s details will appear on the screen

❖ WHEN I click the Edit button

❖ THEN a form page pops up where I can fill in and update the order details

❖ WHEN I click the Save button

❖ THEN Errors will show next to fields that were not filled properly or required fields that were left empty

❖ WHEN I see the errors

❖ THEN I correct the filled in information in the necessary fields

❖ WHEN I click the save button

❖ THEN the view page of the order’s details will appear on screen with the updated details

## System (Feature) Tests:

### First User Story – Admin can create an order and save it:

Happy Path:

An order will be created and saved to the database (checks the store method), testing that the order will be stored and added in the database, that the rerouting is happening and that there were no errors occurring.

Unhappy Path:

An order will not be created and stored because the values are wrong or missing, in the test I am checking if a field has missing required data that the order will not be created so not saved to the database, and that the sessions detects that there are errors.

### Second User Story – Admin can edit an order and save it:

Happy Path:

An order will be updated and saved to the database (checks the update method), testing that the attributes of an existing order changed accordingly and saved properly in the database.

Unhappy Path:

An order will not be updated because the values are wrong or missing, in the test I am making a value that is required and not nullable to null, the test should detect that there is an error, and the field will not be updated.

## Unit tests:

### First User Story – Admin can create an order and save it:

* Checks the automatic status change function according to the attributes (testing event listeners):
* If a production date, a machine, or materials are missing when creating an order the status will be – Admin Hold
* If all the criteria above were set when creating the order but the quality (initial) check is missing (wasn’t created yet) the status will be – Quality Check Pending
* Get the correct view if the admin selected the order:
* If one of the orders has the attribute of “selected” in 1 the isSelected() method will return that order view page
* Only 1 order can be selected at a time (testing the business logic).
* Get the correct view if the admin unselected the order or no order is selected:
* If all of the orders has the attribute of “selected” in 0 the isSelected() method will return null
* If an order has the attribute of “selected” in 1 and the method of unselectOrder is called on that order the “selected” attribute will change to 0, and will be redirected to the index page of all the orders

### Second User Story – Admin can edit an order and save it:

* Checks the automatic status change function according to the attributes:
* If a production date, a machine, or materials are removed the status will be – Admin Hold
* If the order has a check and has all the required attributes (date, machine, and materials) the status of the order will be – Production Pending (this is tested only here because the check is created after the order was created).
* If an order is canceled its status should change to canceled (checks the method of cancelOrder()) and that the user is redirected to the show page of that order.

# Completed tests:

## Automatic testing in GitHub:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text

Description automatically generated

## Testing in terminal:

Text

Description automatically generated

# Evaluation:

## Mistakes Errors that can be detected

My test can detect if there are errors in the event listeners business logic for example if the wrong status will appear to an order which does not match its values. If multiple orders are selected at once (something that shouldn’t be possible), if the redirecting of the view pages is don’t incorrectly, for example if an order is selected but instead of going to the order details page will go to the index page. Any problems with the store or update methods of orders, for example if an order isnt saved when it is supposed to be or saved when its not supposed to for example if the values are not matching the requirements.

## Mistakes Errors that cannot be detected

Reverting the canceled status isnt checked the adding of materials to the order using a store or update method (the two-level form isnt being checked). The limitation to only admin being able to store or update an order (the user is not checked) so if a production worker can create or update an order. Editing or updating of the quality check, and any procedures done by other roles such as production. My tests as well don’t check if the correct side menu with the right details is loaded according to if an order is selected or not.

## Conclusion

My tests didn’t check everything that an administrator can do so there are limitations and possible errors that are not being detected (see above). However, the tests that are running show that the event listeners are working correctly and change the status appropriately, the business logic I tested for both canceling and selecting/unselecting an order work properly, only one order can be selected at a time and the order will become canceled. The app detects and sends out error messages if the fields are filled with incorrect data type or info and if there is required data missing and will catch it in time to prevent creating or updating an order incorrectly. And the rerouting that was tested is working as intended for example if an order is selected its detail’s view page will be shown. Because of all of this I believe my tests show that the implementation of the two user stories I am checking does work properly.