Test Plan

Dogclothes App

Version 1.0

Content

[Test Methods 2](#_Toc119306967)

[Sprint 1: 2](#_Toc119306968)

[Requirements for testing: 2](#_Toc119306969)

[1. The user can save products to the database. 2](#_Toc119306970)

[2. The user can see all the products that are saved in the database. 2](#_Toc119306971)

[3. The user can edit the information of a selected product. 2](#_Toc119306972)

[4. User can delete the selected product. 2](#_Toc119306973)

[Sprint 2 3](#_Toc119306974)

[Requirements for testing: 3](#_Toc119306975)

[5. The user can store the manufacturer information for the product. 3](#_Toc119306976)

[6. The user can see all manufacturers. 3](#_Toc119306977)

[7. The user can delete the manufacturer information. 3](#_Toc119306978)

[8. The user can add a new manufacturer without the product information. 3](#_Toc119306979)

[9. The user can see all products of a specific manufacturer. 3](#_Toc119306980)

[10. The system can get all the products stored in the system in JSON format via REST interface. 3](#_Toc119306981)

[13. The system has a validation check for the addition of a new product (see list number 1). At least the name and type of the product must be included in the product data. 3](#_Toc119306982)

[Features not to be tested 4](#_Toc119306983)

# Test Methods

The functionality of the system will be tested by the programmers at the end of every sprint using J-Unit 5 and Postman. They will use the user requirements to ensure that the pages are working as intended.

Manual testing will be performed during development to ensure that the Thymeleaf-pages work as intended. REST services will be tested manually using postman

## Sprint 1:

### Requirements for testing:

-Product.java and ProductController.java -classes have been created  
-Thymeleaf-pages productlist.html and addproduct.html have been created  
-the tester has access to the project via GitHub  
-test cases and test data need to be ready for testing

### 1. The user can save products to the database.

### 2. The user can see all the products that are saved in the database.

### 3. The user can edit the information of a selected product.

### 4. User can delete the selected product.

## Sprint 2

### Requirements for testing:

-Manufacturer.java and ManufdacturerController.java -classes have been created  
-Thymeleaf-pages manufacturerlist.html and addmanufacturer.html have been created

-REST-services have been implemented  
-the tester has access to the project via GitHub  
-test cases and test data need to be ready for testing

### 5. The user can store the manufacturer information for the product.

User can see the addManufacturer.html page

User can fill in the form and press save

After pressing save user is redirected to manufacturerlist.html and the new manufacturer has been added to the list.

### 6. The user can see all manufacturers.

Endpoint http://localhost:8080/manufacturers returns the page manufacturerList.html

The user can see all the manufacturers

### 7. The user can delete the manufacturer information.

### 8. The user can add a new manufacturer without the product information.

### 9. The user can see all products of a specific manufacturer.

### 10. The system can get all the products stored in the system in JSON format via REST interface.

### 13. The system has a validation check for the addition of a new product (see list number 1). At least the name and type of the product must be included in the product data.

If user tries to save a product without filling in the name or type field, they will get an error message.

# Features not to be tested

The database will not be Stress or Load Tested during sprints 1-2, for it will only be used locally by one user at a time.