

TASK 1 - Writing test cases.

We are about to release our new model of automatic coffee machine for companies. It's aim features are:

- You can push a capsule or fill the ground coffee tank before each service to use the kind you want
- You can select 3 different sizes
- You can pour just hot water
- It goes into standby mode after 20 mins of no use

Write the test cases you would execute to test the software that controls all of these options. No need to detail the steps, just the test case name/scenario.

TASK 1- Solution:

Preconditions:

- Requirements for the automatic coffee machine are checked.
- Automatic coffee machine is connected to electricity.
- There are all necessary buttons on the automatic coffee machine interface.
- There is a water tank.
- There is a container for capsules.
- There is a container for ground coffee.
- The cup holder dimension suitable for each size of coffee cup (small, medium, large).
- It is possible to turn the automatic coffee machine on.
- There is water in the water tank.
-

Ability to prepare capsule or ground coffee.

- A capsule fits into the capsule container.
- Ability to fill a ground coffee container with ground coffee.
- Ability to prepare any size of capsule coffee.
- Ability to prepare any size of ground coffee.
- Ability to prepare any size of ground coffee when there is no water in the water tank.
- Ability to prepare any size of capsule coffee when there is no water in the water tank.
- Ability to prepare any size of coffee when there is no ground coffee in the container.
- Ability to prepare any size of coffee when there is no capsule in the container.
- Ability to prepare any size of coffee when there is nether ground nor capsule coffee in the container.
- Ability to prepare any size of coffee when the water quantity is less than its limit.
- Ability to prepare any size of coffee when the water quantity is more than its limit.
- Ability to prepare any size of coffee when the ground coffee quantity is less than its limit.

- Ability to prepare any size of coffee when the ground coffee quantity is more than its limit.

Ability to select 3 different size of coffee.

- Ability to prepare a small coffee using the capsule option.
- Ability to prepare a medium coffee using the capsule option
- Ability to prepare a large coffee using the capsule option
- Ability to prepare a small coffee using the ground coffee option
- Ability to prepare a medium coffee using the ground coffee option
- Ability to prepare a large coffee using the ground coffee option.

Ability to pour just hot water

- Check an operation of hot water pouring

Standby mode

- The automatic coffee machine goes into standby mode after 20 mins of no use.
- Ability to prepare any size of coffee when the coffee machine is on standby mode.

TASK 2 - Reporting bugs

While testing the machine, you discover that after 20 minutes it actually brews a short espresso instead of going into standby mode. Please write down the bug report as you would file it in a ticket.

TASK 2 - Solution:

Bug report #1

Summary

Automatic coffee machine: Brews short espresso after 20 mins of no use.

Steps to reproduce

1. Turn an automatic coffee machine on .
2. Wait for 20 mins with no action.

Expected result

Automatic coffee machine goes into standby mode after 20 mins of no use.

Actual result

Automatic coffee machine brews short espresso after 20 mins of no use.

Device

Automatic coffee machine

Reproducibility

5/5

Severity

Major

Priority

Medium (Depends on business preferences)

TASK 3 - Exploratory testing

Our business has grown so much, now we also supply our own coffee beans. The problem is that we have [an ecommerce site](#) built for us but we don't own the code and did not have the opportunity to design the requirements.

The product manager asks you to perform some exploratory testing on it and write down your conclusions:

- What sections did you test?
- What parts of our site have the most risk for our business?
- What parts are not that critical?
- What steps would you include in an end to end automated test, meant to run after each deployment, to ensure nothing vital was broken?

TASK 3 - Solution:

- *What sections did you test?*

Home page

Registration form

Log in

Social links

Search

Products

Personal area

Subscription

Rewords

Shopping cart

Payment (without transactions)

Reviews

Contact information

Customer service form and search

Wholesale application form

- *What parts of our site have the most risk for our business?*

Home page

Products

Shopping cart

Payment

Wholesale application form

User registration and login

- *What parts are not that critical?*

Social links

Search

Personal area

Subscription

Rewords

Reviews

Contact information

Customer service form and search

Shopping cart

Shopping cart allows two different clients to add and purchase the only remaining item in stock at the same time.

It doesn't save added items for a registered user after changing to another device.

Review

Review section doesn't work at all. It doesn't directly affect the purchasing process, but it reflects poorly on the image and trustworthiness of the company.

It's impossible to sort reviews by any criteria.

It doesn't let the user go to following pages of reviews.

It doesn't let the user share or rate a review.

It doesn't show any photos.

It doesn't let the user write a review.

Products

There is poor technical information for some items, for example: "M21 Junior Espresso Machines | La Cimbali D".

Missing pictures for some products .

Home page

Subscribe button overlaps email form on the Home page

Blog

Missing some pictures in a Blog section (pages 7-9)

- *What steps would you include in an end to end automated test, meant to run after each deployment, to ensure nothing vital was broken?*

Regression testing:

- New user open home page url
- Click on the account icon
- Click "create account"
- Fill registration form
- Click "Create" button
- Log in
- Search for some item
- Add it to the shopping cart

- Go to the shopping cart
- Click “check out”
- Fill shipping (billing) information
- Click “Pay now”

TASK 4- Writing code

[An API of ice and fire](#) is a free-to-use API that holds a wiki for the Game of Thrones books. For this task, you need to create a project that makes some requests to it.

You can make it as simple or as complicated as you feel comfortable about. It can be a Postman collection or a coding project in the language you desire, just make sure to include a readme file explaining what technology did you choose. What we want to assess with this task is mainly your coding skills, so there is no right or wrong answer, it's just to get a general idea of where you are standing.

https://github.com/yuliazes/wefox_test_tasks.git