# Exercise: Software Quality Assurance Introduction

This document defines the exercises and homework assignments for the [**"QA Basics" Course @SoftUni**](https://softuni.bg/opencourses/qa-basics).

Fill the provided **MS Word document template**. Put the **solution** for each exercise in the document. Name your document QA-Intro-Homework-FirstName-LastName.docx. Submit this document as your homework.

## Think Testing: Gas Station

A woman goes to a gas station and fills fuel in her car. She pays and tries to start the car, but **the car fails to start**. List possible **reasons** that you can think of.

## Think Testing: Tooth Brushing

It is time to teach your 6-year-old child to brush its teeth alone. It needs a clear **step-by-step instruction**, so list the steps and be as detailed as you can.

## Think Testing: 5 Kg Bag

How would you **test a 5 kg capacity grocery shopping paper bag**? Describe the tests that you could perform.

## Login Form UX Problems

As QA engineers, you should be trained to **find issues in the UI and UX** of the apps under test. Given an e-commerce web site "**My Wonderful Shop**", find the **problems** in the below **login form**:

Graphical user interface

Description automatically generated

## Weather Forecast Bug

Your software company works on a project, which displays the **weather forecast** for the next few days. Developers take the weather forecast from **external source**, where the forecast data is given in computer-readable format. The received **weather forecast data** holds weather information for the next 5 days. For each day the following data is obtained: weather icon (sunny / cloudy / light rain / heavy rain / snow / fog / etc.), the min and max temperature, precipitation percentage and wind strength.

During the testing you, as a QA engineer, find out that the **temperatures** come from the external source in **degrees Fahrenheit**, but are displayed in **degrees Celsius without a conversion**. This obviously produces wrong results, such as extremely hot temperatures (like 78 °C). Describe the problem in detail in the table below:

|  |  |
| --- | --- |
| **Mistake** | The developer made the following mistake: … |
| **Bug (location)** | The bug in the code is in the module / function, responsible for: … |
| **Failure (symptoms)** | When the buggy code goes in production, it fails as follows: … |

## Age Checking Machine

As a QA engineer you often will be assigned to **check if certain business logic is correct**. Your company develops a special machine (hardware + software), designed to be put at the entrance of a **bar**, where people come for a drink. The machine should **check the age** of people when they try to enter and tell them, if they are **allowed to walk in**.

You are assigned to **test the machine**, which reads the personal ID card, extracts the **age of the person** from it (integer number), and displays appropriate **message** to each visitor, **based on the age**. The machine **opens the door** when the person is allowed to enter the bar.

By design the machine should implement the **following logic**:

|  |
| --- |
| **Check visitor’s age and print appropriate message** |
| 1. If **age > 0**, and **age < 18**, then **print** "*You are too young to visit our bar*". The door stays closed. 2. If **age > 18**, then **print** "*Welcome to our bar. Enjoy!*" and the door opens. 3. **Otherwise**, **print** "*Invalid age. Please try again*". |

Do you find **any issues** in the above logic?

* What is the **mistake (error)** in the above logic?
* How do we call the situation when we have a **wrong logic in the code**? Name it.
* What will be the **result**, if we **run the code**, which implements the wrong logic?

## Testing an Electric Water Kettle

You are assigned to **test** a simple electrical **water kettle**. The water kettle consists of two parts:

* Electric **water kettle** with 1.0-liter capacity (1500 watts of power)
* **Power base**, used to plug-in the kettle cordlessly

 

The kettle has two buttons:

* **Mechanical button** to **open the lid**. The lid is closed by hand, without any buttons.
* **Switch on/off button** to start boiling the water.

When the kettle is **plugged-in to the power base**, and the power base is **plugged-in** the electrical network, and the kettle holds **enough water** (at least 0.2 liters), the kettle starts heating the water.

* It takes 2-3 minutes for the water to get hot. When the **water starts to boil**, the **kettle automatically stops** (its start button gets into "**off**" state).
* When the water is **hot** and the boiler is **switched on**, it **automatically switches off** after 0.5-2 seconds.
* When the kettle is **empty** or almost empty (holding less than 0.2 liters) and the kettle is **switched on**, it **automatically switches off** after 0.5-2 seconds.

Your task is to think about the **test scenarios** and **test cases** to test the electrical water kettle. Describe the **test scenarios** and each **test case** in the following format:

### Test Scenario: Boil Water

|  |  |
| --- | --- |
| Test case | **Boil 1 liter of water 🡪 success** |
| Description | Pour 1 liter of water, start the kettle, and wait until it gets hot. |
| Steps | 1. Fill 1 liter of cold water in the kettle and close the boiler lid. 2. Plug the power base in the electrical network. 3. Plug the boiler into the power base. 4. Switch on the kettle. 5. Wait until the water gets hot and the kettle automatically switches off (2-3 minutes). |
| Expected results | The boiling process should complete in less than 4 minutes.  The water should get hot.  The kettle should automatically power off when the water gets too hot.  The kettle lid should stay closed. |

|  |  |
| --- | --- |
| Test case | **Boil an empty kettle 🡪 fail** |
| Description | Try to boil an empty kettle (no water inside) and make sure the boiling stops (automatically switches off) almost immediately after starting. |
| Steps | … |
| Expected results | … |

### Test Scenario: Lid Test

|  |  |
| --- | --- |
| Test case | **Open the lid** |

|  |  |
| --- | --- |
| Test case | **Close the lid** |

|  |  |
| --- | --- |
| Test case | **…** |

## Testing a Coffee Machine

As a QA engineer, you are assigned to **test** a simple electrical **espresso** **coffee maker machine**:



**Power ON/OFF**

**Water container**

**Brew short coffee button**

**Brew long coffee button**

**Hot water indicator**

**Coffee outlet**

The coffee machine consists of **machine body**, **water container** (holding water), **coffee outlet** (holding ground coffee blend), **hot water light indicator** and **3 buttons**:

**"Power ON / Power OFF" button**, which switches the machine **on** and **off**.

* When the machine is **powered on**, it automatically **heats the water** **enough** to be ready to brew a coffee. The water stays hot until the machine is switched off.
* Depending on the internal water temperature, the water heating process could take from 5-10 seconds to 1-2 minutes.
* When the water in the machine is **hot enough** to brew a coffee, the "**hot water indicator light**" is **on**. Otherwise, the light is **off**.
* If the **water container is empty**, the machine **starts beeping** (on intervals of 10 seconds, until powered off or until enough water is filled inside the container).

**"Brew" buttons**: brew a **short coffee** (60 ml) or a **long coffee** (120 ml)



* When a **"brew" button** is pressed, if the water is **hot enough**, and the **coffee outlet is put correctly**, the machine starts making coffee (it slowly pours hot water through the coffee blend into the coffee cup below):



* The **brew process stops** when the coffee is **ready** (after successfully pouring the intended amount of hot water, typically in 20-40 seconds).
* If some of the **"brew"** buttons is pressed during the brew operation, its stops (unfinished).
* After the brew process is finished (or stopped), **if the water container is empty**, the machine **starts beeping** (as described above).

Your task is to think about the **test scenarios** and **test cases** to test the coffee machine. How many test cases do you need to fully test the coffee machine? Describe the **test scenarios** and each **test case** in the following format:

### Test Scenario: Brew a Coffee

|  |  |
| --- | --- |
| Test case | **Brew a small coffee 🡪 success** |
| Description | Start the coffee machine, put water, put ground coffee in the outlet, and brew a cup of coffee. |
| Steps | 1. Power on the machine. 2. Put ground coffee blend in the coffee outlet. 3. Fill the water container to its max level. 4. Wait until the "hot water" indicator lights up. 5. Put an empty coffee cup under the coffee outlet. 6. Press the "brew small coffee" button. 7. Wait until the brew process finishes. |
| Expected results | The brew process should complete in less than 50 seconds.  The coffee cup should hold a hot small coffee (60 ml).  The machine should stay powered on.  The "hot water" indicator light could be on or off (both states are correct).  The machine should have enough water in its water container (it should not beep). |

|  |  |
| --- | --- |
| Test case | **Brew a coffee with no water 🡪 fail** |
| Description | Start the coffee machine, empty the water container, try to brew a cup of coffee, expect the coffee machine to start beeping to indicate that the water is not enough. |
| Steps | … |
| Expected results | … |

|  |  |
| --- | --- |
| Test case | **Brew a coffee long coffee 🡪 success** |

|  |  |
| --- | --- |
| Test case | **…** |

### Test Scenario: Machine On / Off

|  |  |
| --- | --- |
| Test case | **Switch off 🡪 check light indicator** |

|  |  |
| --- | --- |
| Test case | **Switch on with no water 🡪 beeping** |

|  |  |
| --- | --- |
| Test case | **…** |

Think about the unusual cases, e.g., "*try to brew coffee when the water is not enough*", "*try to brew a coffee when the water is not hot*", etc.