# Homework: Test Levels and Test Types

## Unit Testing in the Real Life: Testing a Battery

|  |  |
| --- | --- |
| **Test #1** | Take a bulb 1.5V and check if the batter works as expected: the bulb should light up after connection properly. |
| **Test #2** | Take multimeter and check the **voltage**.   * It should = 1.5v. |
| **Test #3** | Take the battery and check it visually:   * Check its **length**. * Check its **diameter**. * Check if is has a from **cylinder**. * Check for leakage , corrosion, etc. |
| **Test #4** | Check with a compatible **flashlight**. This will check two things:  Whether battery size matches the flashlight.  Whether the batteries work as expected (light the bulb). |
| **Test #5** | Check the **labels** on the battery:   * The denoted size should be “AA”. * The denoted voltage should “1.5V”. |
| **Test #6** | Check if “+” and “-“ are correctly positioned. Se a multimeter. |
| **Test #7** | Environmental test:   * Low temperature, e.g. 2 degree Celsius. * High temperature, e.g. 45 degree Celsius. |
| **Test #8** | Check the expiration data label. It should be in the future. |
| **Test #9** | Check |

## Unit Testing in the Real Life: Testing a Light Bulb

|  |  |
| --- | --- |
| **Test #1** | Visual test:   * We check whether the shape and size meet the specified standard. * Is the glass strong? * is the wire inside healthy? |
| **Test #2** | Functional test:   * Can a fitting be fitted to the correct size? * Whether the thread is wound correctly. * We test with a flashlight or other lighting device. |
| **Test #3** | Maximum and minimum light test:   * We test with different batteries with a minimum voltage how much it will light up and a battery with a maximum voltage until how much it will light up and when it will burn out. |
| **Test #4** | Overheat Test:   * When connected to a battery for a long time, how long can it work and how much will it heat up. |
| **Test #5** | Extreme Temperature Test:   * At -10 degrees Celsius. * At 45 degrees Celsius. |
| **Test #6** | Check for inscriptions on for voltage and model. |
| **Test #7** |  |
| **Test #8** |  |
| **Test #9** |  |
| **Test #10** |  |

## Unit Testing in the Software World: Age Checker

|  |  |  |
| --- | --- | --- |
| **#** | **Test Description** | **Pass / Fail** |
| **Test #1** | * AgeChecker(5) 🡪 child * AgeChecker(12.9) 🡪 child * AgeChecker(13) 🡪 teenager * AgeChecker(19.5) 🡪 teenager * AgeChecker(20) 🡪 adult * AgeChecker(21) 🡪 adult * AgeChecker(50) 🡪 adult * AgeChecker(64.7) 🡪 adult * AgeChecker(65) 🡪 elder * AgeChecker(75.3) 🡪 elder * AgeChecker(95) 🡪 elder * AgeChecker(150) 🡪 elder * AgeChecker(-5) 🡪 error * AgeChecker(155) 🡪 error * AgeChecker(1222) 🡪 error | Pass |

## Unit Testing in the Software World: Income Checker

|  |  |  |
| --- | --- | --- |
| **#** | **Test Description** | **Pass / Fail** |
| **Test #1** | 1.Test the basic functionality:  Test Case 1:  Input: 500  Expected Output: “low”  Test Result: Passed  Test Case 2:  Input: 1500  Expected Output: “mid”  Test Result: Passed  Test Case 3:  Input: 3500  Expected Output: “high”  Test Result: Passed  Test Case 4:  Input: -500  Expected Output: “error”  Test Result: Passed  Test Case 5:  Input: 1000  Expected Output: “mid”  Test Result: Passed  Test Case 6:  Input: 2999  Expected Output: “mid”  Test Result: Passed  Test Case 7:  Input: 3000  Expected Output: “high”  Test Result: Passed  Test Case 8:  Input: 0  Expected Output: “low”  Test Result: Passed  2.Test edge cases:   * I would test the function with inputs that are on the boundaries of the categories, such as 1000 and 3000, to verify that the function correctly categorizes these inputs.   3.Test with invalid inputs:   * I would test the function with inputs that are not valid monthly incomes, such as non-numeric inputs or very large inputs, to ensure that the function handles such inputs gracefully and returns an appropriate error message.   4.Test with different locales:   * If the function is meant to be used in multiple locales, I would test it with inputs in different currencies and with different decimal separators to verify that it works correctly in all locales. | Pass |

## Integration Testing in the Real Life: Lighting the Bulb

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Test #1** | Implement the following circuit, using the provided components:  The bulb should light. | | **Test #2** | Implement the following circuit, using the provided components:  **Switch on** the switch button 🡪 the bulb should light. | | **Test #3** | Implement the following circuit, using the provided components:  **Switch off** the switch button 🡪 the bulb should **not light**. | | **Test #4** |  | | **Test #5** |  | | |  |  | | --- | --- | | **Test #1** | Implement the following circuit, using the provided components:  The bulb should light. | | **Test #2** | Implement the following circuit, using the provided components:  **Switch on** the switch button 🡪 the bulb should light. | | **Test #3** | Implement the following circuit, using the provided components:  **Switch off** the switch button 🡪 the bulb should **not light**. | | **Test #4** |  | | **Test #5** |  | |
|  |  |
| |  |  | | --- | --- | | **Test #1** | Implement the following circuit, using the provided components:  The bulb should light. | | **Test #2** | Implement the following circuit, using the provided components:  **Switch on** the switch button 🡪 the bulb should light. | | **Test #3** | Implement the following circuit, using the provided components:  **Switch off** the switch button 🡪 the bulb should **not light**. | | **Test #4** |  | | **Test #5** |  | | |  |  | | --- | --- | | **Test #1** | Implement the following circuit, using the provided components:  The bulb should light. | | **Test #2** | Implement the following circuit, using the provided components:  **Switch on** the switch button 🡪 the bulb should light. | | **Test #3** | Implement the following circuit, using the provided components:  **Switch off** the switch button 🡪 the bulb should **not light**. | | **Test #4** |  | | **Test #5** |  | |
|  |  |
| |  |  | | --- | --- | | **Test #1** | Implement the following circuit, using the provided components:  The bulb should light. | | **Test #2** | Implement the following circuit, using the provided components:  **Switch on** the switch button 🡪 the bulb should light. | | **Test #3** | Implement the following circuit, using the provided components:  **Switch off** the switch button 🡪 the bulb should **not light**. | | **Test #4** |  | | **Test #5** |  | | |  |  | | --- | --- | | **Test #1** | Implement the following circuit, using the provided components:  The bulb should light. | | **Test #2** | Implement the following circuit, using the provided components:  **Switch on** the switch button 🡪 the bulb should light. | | **Test #3** | Implement the following circuit, using the provided components:  **Switch off** the switch button 🡪 the bulb should **not light**. | | **Test #4** |  | | **Test #5** |  | |

## \* Integration Testing in the Software World: Ads

|  |  |
| --- | --- |
| **Test #1** | We test the login form, whether it opens the correct page with the correct entry form. |
| **Test #2** | When all functions must switch to next page with right form to login.  After a successful login, the main page should have the user's name written on it and load several additional options. |
| **Test #3** | On logout, it should successfully load the home page without the user's data. otherwise there is a bug in the system and personal data may be lost. |
| **Test #4** | Test of the registration button, during normal operation of the application, it should load a page with the registration form, the name, password and e-mail data. After which there will be a verification of the e-mail. |
| **Test #5** |  |
| **Test #6** |  |
| **Test #7** |  |
| **Test #8** |  |

## \* Integration Testing in the Software World: Credit Risk

|  |  |
| --- | --- |
| **Tests** | The following 12 tests cover each combination of age + income:   * CreditRisk(age: 5, income: 500) 🡪 100% * CreditRisk(age: 6, income: 2000) 🡪 100% * CreditRisk(age: 7, income: 6000) 🡪 100% * CreditRisk(age: 15, income: 700) 🡪 80% * CreditRisk(age: 15, income: 2000) 🡪 72% * CreditRisk(age: 17, income: 6700) 🡪 64% * … * … * …   Additional tests for invalid input:   * CreditRisk(age: -5, income: 500) 🡪 error * CreditRisk(age: 6, income: -2000) 🡪 error   Regression test:   * CreditRisk(age: 17, income: 0) 🡪 80%   + Bug in the sample calculator: <http://softuni-qa-loadbalancer-2137572849.eu-north-1.elb.amazonaws.com/credit-risk/> * CreditRisk(age: 0, income: 1000) 🡪 100%   + Bug in the sample calculator: <http://softuni-qa-loadbalancer-2137572849.eu-north-1.elb.amazonaws.com/credit-risk/> * CreditRisk(age: 0, income: 0) 🡪 100% |

Another solution (or way of thinking):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | child | teenager | adult | elder | negative |
| low | 100% |  |  |  |  |
| mid | 100% |  |  |  |  |
| high | 100% |  |  |  |  |
| negative | error | error | error | error | error |

I also found bug in the credit risk calculator: <http://softuni-qa-loadbalancer-2137572849.eu-north-1.elb.amazonaws.com/credit-risk/>

* When the age or income holds “0”, the messages under the “age” and “incomes” boxes are incorrect:

Diagram

Description automatically generated with low confidence

* There is a **UI bug**. When the screen is not big enough, the downside of the form gets cut. Also when we zoom-in / zoom-out, content may become missing:

Graphical user interface, text

Description automatically generated

* …

## System Testing in the Real Life: Flashlight

|  |  |
| --- | --- |
| **Test #1** | Test switch on / switch off the light.  We take the flashlight. Put new batteries correctly. Switch on the flashlight 🡪 the bulb should light. Switch off the flashlight 🡪 the bulb should light off. |
| **Test #2** | Test battery replacement. |
| **Test #3** | Test bulb replacement. |
| **Test #4** | Test battery duration. At least 1 hour of lighting with new batteries. |
| **Test #5** | Test the illumination distance. It should illuminate cleanly at distance of 30 meters or less (with new batteries). |
| **Test #6** | Shock resistance test: fall from the table and check if it still works correctly. |
| **Test #7** | Operation under high / low temperature. |
| **Test #8** | Overheat test. |
| **Test #9** | Water resistance test. |
| **Test #10** |  |

## System Testing in the Real Life: Digital Scale

|  |  |
| --- | --- |
| **Test #1** | First test "Normal or initial" - we place a pre-measured weight of 5 kg. therefore, the scale must read the same kilograms to be true.  Second "human weight" test, we choose a person with a normal weight, which is a statistical average, and we check on a second scale whether it is the same weight.  Third test "For maximum weight" we choose a person or an object with the maximum mass that can be placed on the scale to check if it will read.  Fourth test "Over the maximum weight" with a person and an additional weight that exceeds the maximum mass.  Fifth endurance test - we let a person or a heavy object stand for 30 minutes on the scale. After that, we check the accuracy again with the previously recorded results.  Sixth test "Extreme" works when it is not damaged on a flat surface.  Does it work when wet with water?  After falling from a height of 50 cm  We test after 1 hour of exposure to the sun and cool temperatures.  From this test, we also see if the display will hold up in addition to the accuracy of the scale itself. |

## System Testing in the Software World: Number Calculator

|  |  |
| --- | --- |
| **Test #1** | Calc(5, +, 3) 🡪 8  Test passed |
| **Test #2** | Calc(5, +, 0) 🡪 5  Test passed |
| **Test #3** | Calc(Infinity, +, 1) 🡪 Infinity  Test passed |
| **Test #4** | Calc(-Infinity, +, 1) 🡪 Infinity Test passed |
| **Test #5** | Calc(pesho, +, 1) 🡪 invalid input Test passed |
| **Test #6** | Calc(1000000000000, +, 5) 🡪 1000000000005  Test failed! |
| **Test #7** | Calc(5, \*, 3) 🡪 15  Test passed |
| **Test #8** | Calc(15, /, 3) 🡪 5  Test passed |
| **Test #9** | Calc(5, -, 3) 🡪 2  Test passed |
| **Test #10** | Calc(5^%$ , +, 5) 🡪 10  Test failed! |
| **Test #11** |  |
| **Test #12** |  |
| **Test #13** |  |
| **Test #14** |  |

## Acceptance Testing in the Real Life: Flashlight

|  |  |
| --- | --- |
| **Test #1** | The customer takes the flashlight, **switch on / off** the light, and assures it works. |
| **Test #2** | The customer checks the flash **illumination**. |
| **Test #3** | The customer checks how easy it is to **replace the batteries**. |

## Acceptance Testing in the Real Life: Digital Scale

|  |  |
| --- | --- |
| **Test #1** | Visual test:   * Is everything safe on it, inspect for visible damage. |
| **Test #2** | Testing if it is easy to move, like the weight of the scale itself - does it weigh. |
| **Test #3** | Test if it works:   * Once it is on, if you step on it, it will activate and the display will work. |
| **Test #4** |  |
| **Test #5** |  |

## Acceptance Testing in the Software World: Number Calculator

|  |  |
| --- | --- |
| **Test #1** | Calc(5, +, 3) 🡪 8  Test passed |
| **Test #2** | Calc(5, +, 0) 🡪 5  Test passed |
| **Test #3** | Calc(Infinity, +, 1) 🡪 Infinity  Test passed |
| **Test #4** | Calc(-Infinity, +, 1) 🡪 Infinity Test passed |
| **Test #5** | Calc(pesho, +, 1) 🡪 invalid input Test passed |
| **Test #6** | Calc(1000000000000, +, 5) 🡪 1000000000005  Test failed! |
| **Test #7** | Calc(5, \*, 3) 🡪 15  Test passed |
| **Test #8** | Calc(15, /, 3) 🡪 5  Test passed |
| **Test #9** | Calc(5, -, 3) 🡪 2  Test passed |
| **Test #10** | Calc(5^%$ , +, 5) 🡪 10  Test failed! |

## Functional and Non-Functional Tests: Flashlight

|  |  |
| --- | --- |
| **Functional Tests** | **Non-Functional Tests** |
| Test switch on / switch off the light.  We take the flashlight. Put new batteries correctly. Switch on the flashlight 🡪 the bulb should light. Switch off the flashlight 🡪 the bulb should light off. | Shock resistance test: fall from the table and check if it still works correctly. |
| Test battery replacement. | Operation under high / low temperature. |
| Test bulb replacement. | Overheat test. |
| Test battery duration. At least 1 hour of lighting with new batteries. | Water resistance test. |
| Test the illumination distance. It should illuminate cleanly at distance of 30 meters or less (with new batteries). | Can it be used as a nutcracker? |
|  |  |
|  |  |
|  |  |
|  |  |