Hochschule Fulda

University of Applied Sciences

Test-Oriented Software Development

Unit Test Cases and Unit Code

Lloyd M. Dzokoto

Matrikel-Nr: 246985

Winter Semester 2016, Fulda

Table of Contents:

Performed by: Lloyd Dzokoto

Date: 24.03.2017

Venue: Hochschule, Fulda

Remarks: Test cases covers all critical data areas of the Registration Module.

Function: checkForm

Priority: Medium

Design Technique: Syntax Test (Backus-Naur-Form)

<name>$^{50}_{6}$ ::= <letter>“ ”<letter> | <letter>“ ”<letter>“-” | “ ”<letter>
<letter>$^{15}_{2}$::= ‘A’ | ‘B’ | ‘C’| ‘D’ | … |’Z’ | a’ | ‘b’ | ‘c’ | ‘d’ | … | ‘z’

Input datatype : String

Required field: Yes

Justification: Test the functionality of name validation.

Field	Input	Outcome
Fullname	“Jon Black”	valid
	“mike jet-lee”	valid
	“Lee Hu Tan”	valid
	“Tai Mu-Tan”	valid
	“Jet-Li”	invalid
	“Jackie_chan”	invalid
	“12345678”	invalid
	“ ”	invalid

Function: checkForm

Priority: High

Design Technique: Equivalence Class Partition

Input datatype : Integer

Required field: Yes

Justification: To ensure address code is within Fulda.

Field	Equivalence Classes	Representative
AddressCode	vEC_1: 36029 < x < 36051	36037
	iEC_1: x < 36030	5
	iEC_2: x > 36050	77372
	iEC_3: NaN	hi

Method: checkForm

Priority: High

Design Technique: Syntax Test (Backus-Naur-Form)

<passid>$^{8}_{6}$ ::= <letter>{<number>}
<letter>$^{3}_{1}$::= ‘A’ | ‘B’ | ‘C’| ‘D’ | … |’Z’ | ‘a’ | ‘b’ | ‘c’ | ‘d’ | … | ‘z’
<number> ::= 0 | 1| 2| 3 | ...|9

Input datatype : String

Required field: Yes

Justification: To ensure valid passport id formats are given to check if its authentic.

Field	Input	Outcome
PassportId	“G0123456”	valid
	“PAN54785”	valid
	“zAb9345”	valid
	“Th3986”	valid
	“ind3986”	valid
	“k34765P”	invalid
	“F_923764”	invalid
	“G012”	invalid
	“12345678”	invalid
	“ ”	invalid

Precondition: Registrant must have access to:

- a user name and password.
- website link through Internet.

Test Case #	Test Scenario	Test Steps	Test Data	Expected response	Outcome	Pass/Fail
TC01	Check Submit button functionality	1. login 2. enter fullname 3. enter addresscode 4. enter passportid 5. click Submit button	Fullname : “Jon Black” AddressCode: 36041 PassportId: “Ind2345”	Form submitted	Form submitted	pass
TC02			Fullname : “Jon Black” AddressCode: 41 PassportId: “Th3986”	Form submitted	Error: invalid AddressCode	fail
TC03			Fullname : “Tai Mu-Tan” AddressCode: 77372 PassportId: “Th3986”	Form submitted	Error: invalid AddressCode	fail
TC04			Fullname : “Tai Mu-Tan” AddressCode: hi PassportId: “ind3986”	Error: wrong data type	Error: wrong data type	pass
TC05			Fullname : “Tai Mu-Tan” AddressCode: 36045 PassportId: “F_923764”	Form submitted	Error: invalid PassportId	fail
TC06			Fullname : “mike jet-lee” AddressCode: 36041 PassportId: “G012”	Form submitted	Error: invalid PassportId	fail
TC07			Fullname : “Arthur Knight” AddressCode: 36033 PassportId: “12345678”	Form submitted	Error: invalid PassportId	fail

TC08			Fullname : “Kevin Schmidt” AddressCode: 36030 PassportId: “ind3986T”	Form submitted	Error: invalid PassportId	fail
TC09			Fullname : “Uwe” AddressCode: 36034 PassportId: “zAb934”	Form submitted	Error: invalid fullname	fail
TC10			Fullname : “ ” AddressCode: “” PassportId: “ ”	Error:required : Fullname AddressCode PassportId	Error:required : Fullname AddressCode PassportId	pass

Postcondition: Processing continues to check passport id and if valid, stores data in database else registrant receives unsuccessful registration notice per post.

Code before Static Review:

```
function checkForm(form)
{
    var name_error = document.getElementById("name_error");
    var zip_error = document.getElementById("zip_error");
    var passid_error = document.getElementById("passid_error");
    var name_reg = /^[A-Za-z]+\s[A-Za-z]+\s*[A-Za-z]*$|[A-Za-z]+\s[A-Za-z]+\-[A-Za-z]*$/;
    var passid_reg = /^[A-Za-z]{1}[0-9]{5,7}$|^[A-Za-z]{2}[0-9]{4,7}$|^[A-Za-z]{3}[0-9]{3,7}$/;


    if (form.name.value.length === 0) {
        processError(form.name, name_error, "Fullname is required");
        return false;
    }

    if (form.zip.value.length === 0) {
        processError(form.zip, zip_error, "ZipCode is required");
        return false;

    }
    if (form.passid.value.length === 0) {
        processError(form.passid, passid_error, "Passport Id is required");
        return false;
    }
    if (!form.name.value.match(name_reg)) {
        processError(form.name, name_error, "Fullname has wrong format");
        return false;
    }
    if (form.zip.value < 36030 | form.zip.value > 36050) {
        processError(form.zip, zip_error, "Invalid Zipcode");
        return false;

    }
    if (!form.passid.value.match(passid_reg)) {
        processError(form.passid, passid_error, "Passport Id has wrong format");
        return false;
    }
    return true;
}
function processError(field, field_error, field_msg) {

    field.style.border = "1px solid red";
    field_error.textContent = field_msg;
    field.focus();

    return false;
}
```

Code after Static Review changes:

```
function checkForm(form)
{
    var name_error = document.getElementById("name_error");
    var zip_error = document.getElementById("zip_error");
    var passid_error = document.getElementById("passid_error");

    /* regular expressions for name and passport id formats*/
    var name_reg = /^[A-Za-z]+\s[A-Za-z]+\s*[A-Za-z]*$|[A-Za-z]+\s[A-Za-z]+\-[A-Za-z]*$/;
    var passid_reg = /^[A-Za-z]{1}[0-9]{5,7}$|^[A-Za-z]{2}[0-9]{4,7}$|^[A-Za-z]{3}[0-9]{3,7}$/;


    /* checking for empty string as input */
    if (form.name.value.length === 0) {
        processError(form.name, name_error, "Fullname is required");
        return false;
    }

    /* check if input fullname has right format*/
   if (!form.name.value.match(name_reg)) {
        processError(form.name, name_error, "Fullname has wrong format");
        return false;
    }

    /* checking for empty integer as input */
    if (form.zip.value.length === 0) {
        processError(form.zip, zip_error, "ZipCode is required");
        return false;

    }
    /* checking for empty string as input */
    if (form.passid.value.length === 0) {
        processError(form.passid, passid_error, "Passport Id is required");
        return false;
    }

    /* check if addres code is within Fulda*/
    if (form.zip.value < 36030 | form.zip.value > 36050) {
        processError(form.zip, zip_error, "Invalid Zipcode");
        return false;

    }

    /* check if input passport id has right format*/
    if (!form.passid.value.match(passid_reg)) {
        processError(form.passid, passid_error, "Passport Id has wrong format");
        return false;
    }
    return true;
}
function processError(field, field_error, field_msg) {

    field.style.border = "1px solid red";
    field_error.textContent = field_msg;
    field.focus();

    return false;
}
```

HTML Form:

```
<!DOCTYPE html>
<html >
    <head>
        <meta charset="UTF-8">
        <title>Fulda Stadt</title>
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-
reset/2.0/reset.min.css">
        <link rel="stylesheet" href="Presentation/css/formstyle.css">
        <script src="Scripts/checkInput.js" type="text/javascript"></script>
</head>

<body>
    <div class="container">
        <div class="login">
            <h1 class="login-heading">
                <strong>Welcome to Fulda!</strong>
            </h1>
            <h2>A Beautiful City</h2>
            <br><br>
            <p>Register below</p>
            <form name="cForm" method="post" action="FuldaStadtServlet" onsubmit="return
checkForm(this)">
                <div>
                    <input type="text" name="name" placeholder="Full name" class="input-txt" />
                    <div id="name_error" class="val_error"></div>
                </div>
                <div>
                    <input type="text" name="email" placeholder="email" class="input-txt" />
                </div>
                <div>
                    <input type="text" name="address" placeholder="Street / Number" class="input-txt" />
                </div>
                <div>
                    <input type="number" name="zip" placeholder="Zip code" class="input-txt"/>
                    <div id="zip_error" class="val_error"></div>
                </div>
                <div>
```

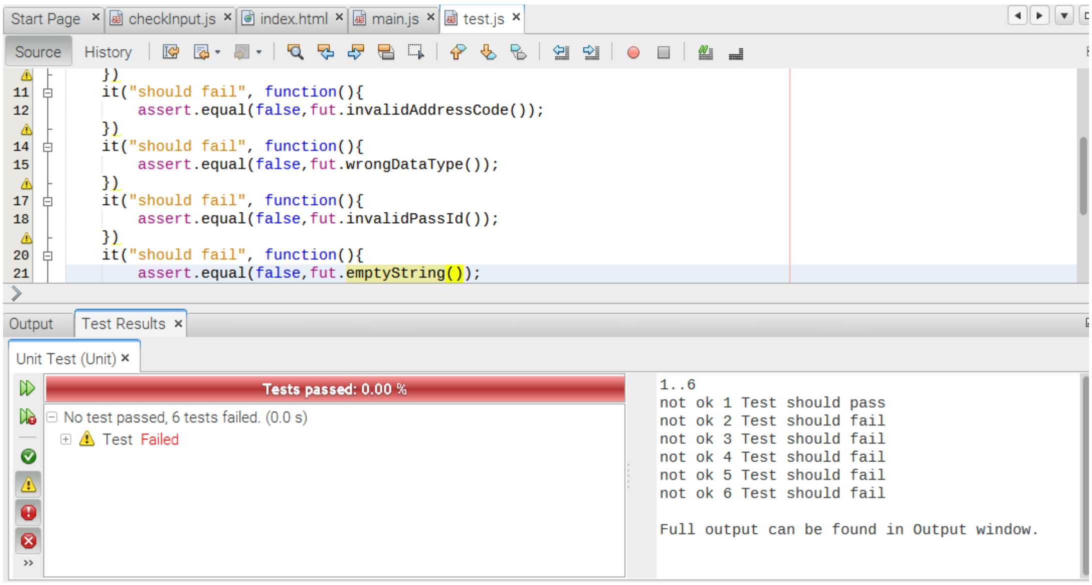
```
                    <input type="text" name="passid" placeholder="Passport Id" class="input-txt" />
                    <div id="passid_error" class="val_error"></div>
                </div>
                <div class="login-footer">
                    <input type="submit" class="btn btn--right">
                    <input type="reset" value="Cancel" class="btn btn--left">
                </div>

            </form>
        </div>
    </div>
```

formstyle.css:

```
@import url(http://fonts.googleapis.com/css?family=Tenor+Sans);
html {
  background-color: #5D92BA;
  font-family: "Tenor Sans", sans-serif;
}

h2 {
    color: white;
}

p {
  font-size: .9em;
  color: white;
}

.container {
  width: 500px;
  height: 400px;
  margin: 0 auto;
}

.login {
  margin-top: 50px;
  width: 450px;
}

.login-heading {
  font: 1.8em/48px "Tenor Sans", sans-serif;
  color: white;
}

.input-txt {
  width: 100%;
  padding: 20px 10px;
  background: #5D92BA;
  border: none;
  font-size: 1em;
  color: white;
  border-bottom: 1px dotted rgba(250, 250, 250, 0.4);
  -moz-box-sizing: border-box;
  -webkit-box-sizing: border-box;
  box-sizing: border-box;
  -moz-transition: background-color 0.5s ease-in-out;
  -o-transition: background-color 0.5s ease-in-out;
  -webkit-transition: background-color 0.5s ease-in-out;
  transition: background-color 0.5s ease-in-out;
}
.input-txt:-moz-placeholder {
  color: #81aac9;
}
.input-txt:-ms-input-placeholder {
  color: #81aac9;
}
.input-txt::-webkit-input-placeholder {
  color: #81aac9;
}
.input-txt:focus {
  background-color: #4478a0;
}

.login-footer {
  margin: 10px 0;
  overlow: hidden;
  float: left;
  width: 100%;
}

.btn {
  padding: 15px 30px;
  border: none;
  background: white;
  color: #5D92BA;
}

.btn--right {
  float: right;
}

.icon {
  display: inline-block;
}

.icon--min {
  font-size: .9em;
}
```

```
.lnk {
  font-size: .8em;
  line-height: 3em;
  color: white;
  text-decoration: none;

  .val_error{
      color: #4478a0;
  }
}
```

Unit Testing with JavaScript (mocha and chai):



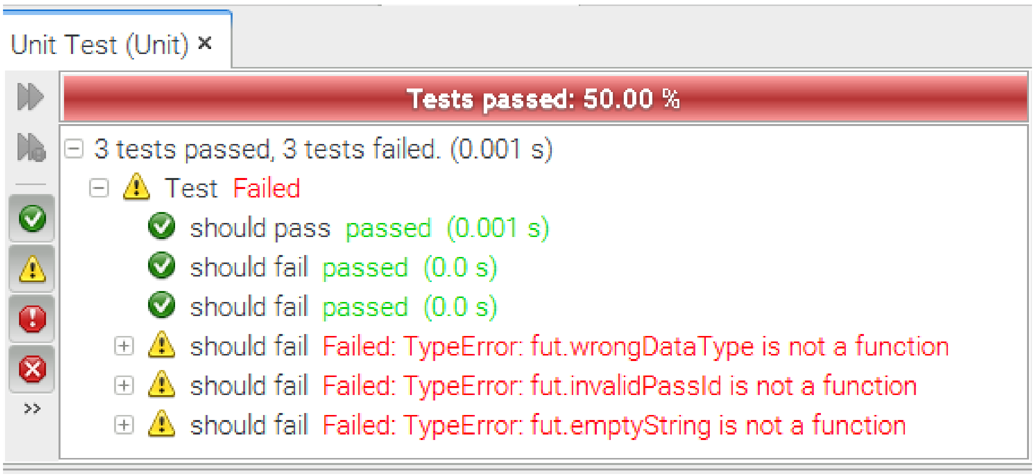
```
})
it("should fail", function(){
    assert.equal(false,fut.invalidAddressCode());
})
it("should fail", function(){
    assert.equal(false,fut.wrongDataType());
})
it("should fail", function(){
    assert.equal(false,fut.invalidPassId());
})
it("should fail", function(){
    assert.equal(false,fut.emptyString());
```

Tests passed: 0.00 %

No test passed, 6 tests failed. (0.0 s)

Test Failed

```
1..6
not ok 1 Test should pass
not ok 2 Test should fail
not ok 3 Test should fail
not ok 4 Test should fail
not ok 5 Test should fail
not ok 6 Test should fail

Full output can be found in Output window.
```



Tests passed: 50.00 %

3 tests passed, 3 tests failed. (0.001 s)

Test Failed

should pass passed (0.001 s)

should fail passed (0.0 s)

should fail passed (0.0 s)

should fail Failed: TypeError: fut.wrongDataType is not a function

should fail Failed: TypeError: fut.invalidPassId is not a function

should fail Failed: TypeError: fut.emptyString is not a function