Test Design Techniques

Bledea Bogdan

The application I have chosen is a personal project, exactly a Single Page Application which people can use to create music loops, save and listen them. The application is called Splicer and it was developed in Vanilla Javascript and Node.js, using SQL Server for Database management.

The test techniques I have chosen to use for testing this application are:

- Function tours,
- Quick tests,
- Scenario-based testing,
- Multivariable testing.

I have use Javascript to implement the test using the Node.js package called selenium-webdriver, mochawesome, chai, chai-as-promise and nyc. The packages can be found at: https://www.npmjs.com/package.

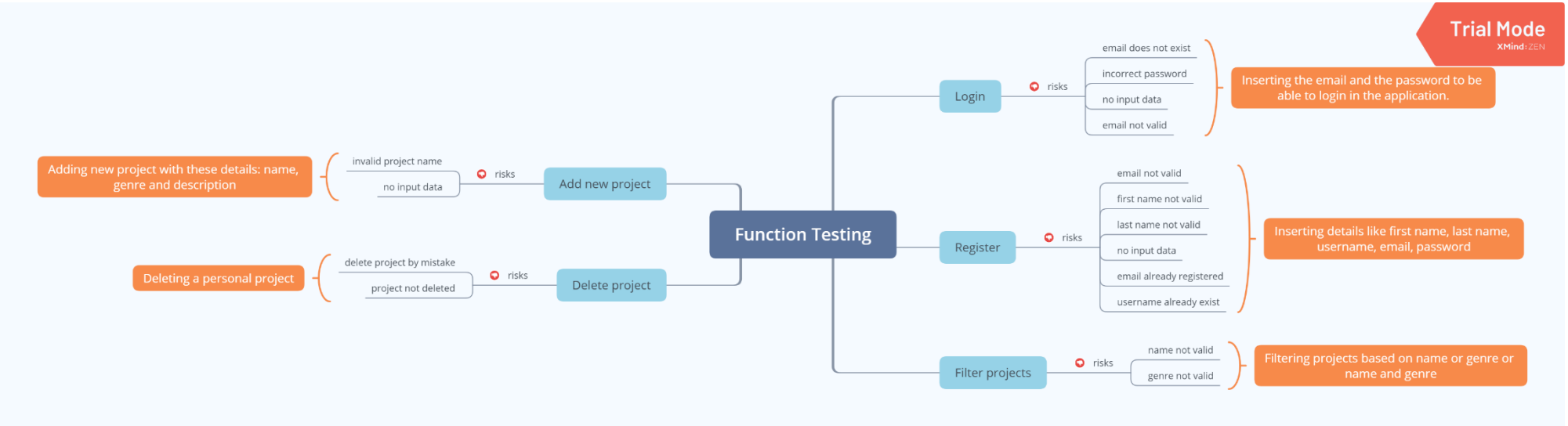
Now, I'm going to describe a little the packages I have used. I will start with the selenium-webdriver, which is a browser automation library that helped me to use for any task of my application that requires automating interaction with the browser.

The second package is mochawesome which is a custom reporter for use with the Javascript testing framework. I used to display in a beautiful way the test I wrote for the application.

Chai is a BDD / TDD assertion library for node and the browser that can be delightfully paired with any javascript testing framework. I used chai-as-promised also because the test I have wrote are asynchronous.

Nyc is a framework I used for test coverage.

TT1 – Function tours



TT2 – Quick tests

I have chosen to test some UX features like buttons clicking, dropdowns appearing, buttons and inputs finding and many more. I have tested the following:

- finding profile logo
- showing the dropdown when profile logo is clicked
- finding profile and logout options
- going to profile page when profile option is clicked
- going to login page when logout option is clicked
- finding the hamburger icon
- closing the sidebar when hamburger icon is clicked
- reopen the sidebar when hamburger icon is clicked two times
- finding the edit button
- open the edit project details page when edit button is clicked

TT3 – Use-case testing

Use case	User authentication
Actor	User
Events	1. User starts the application. 2. System shows `Login` page. 3. User inserts a correct email. 4. User inserts a wrong password. 5. User clicks login button. 6. System validates the data. 7. Systems shows that the password is incorrect.
Input	User sees the `Login` page
Output	User sees the `Login` page
Quality	Validating process does not take more than 2 ms.

Use case	User authentication
Actor	User
Events	1. User starts the application. 2. System shows `Login` page. 3. User inserts a correct email. 4. User inserts a wrong password. 5. User clicks login button. 6. System validates the data. 7. Systems shows that the password is incorrect.
Input	User sees the `Login` page
Output	User sees the `Login` page
Quality	Validating process does not take more than 2 ms.

Use case	User authentication
Actor	User
Events	1. User starts the application. 2. System shows `Login` page. 3. User inserts email and password. 4. User clicks login button. 5. System validates the data. 6. Systems authenticates the user and assigns a token to it. 7. System shows `Projects` page.
Input	User sees the `Login` page
Output	User sees the `Projects` page
Quality	Validating and login process does not take more than 5 ms.

Use case	Add new project
Actor	User
Events	1. User clicks the Add new project. 2. System redirects user to add new project page. 3. User inserts name of project. 4. User selects genre of project. 5. User inserts description of project. 6. User clicks confirm button. 7. System validates the data and the required inputs. 8. System adds new project. 9. System redirects user to `Projects` page.
Input	User is authenticated. User sees the `Projects` page.
Output	User sees the `Projects` page
Quality	Validating process does not take more than 5ms.

Use case	Add new project
Actor	User
Events	1. User clicks the Add new project. 2. System redirects user to add new project page. 3. User selects genre of project. 4. User inserts description of project. 5. User clicks confirm button. 6. System validates the data and the required inputs. 7. System shows that the project name input is required. 8.
Input	User is authenticated. User sees the `Projects` page.
Output	User sees the `Add new project` page
Quality	Validating process does not take more than 5ms.

Use case	Add new project
Actor	User
Events	1. User clicks the Add new project. 2. System redirects user to add new project page. 3. User inserts project name. 4. User inserts description of project. 5. User clicks confirm button. 6. System validates the data and the required inputs. 7. System shows that the genre input is required.
Input	User is authenticated. User sees the `Projects` page.
Output	User sees the `Add new project` page
Quality	Validating process does not take more than 5ms.

The main difference between Use case testing and Scenario based testing is that Scenario based testing uses stories that relies on good scenarios, while the Use case testing uses Sequence diagrams.

TT4 - Multivariable testing

I have not used an online tool to generate values for the variables, but I have used the PICT tool. The files are: Filter.txt and Register.txt.

Inside Filter.txt we can find the following values:

Name: ah - mind 96bpm.mp3, bass 1- swag 114bpm.mp3, asdasda, chord 2 - cruel 67bpm.mp3
Type: Closed Hihat, Fx, Kick, Open Hihat, asdasda

After using the pict cmd we obtain the following table:

Name	Type
asdasda	asdasda
bass 1- swag 114bpm.mp3	Fx
bass 1- swag 114bpm.mp3	asdasda
chord 2 - cruel 67bpm.mp3	Fx
bass 1- swag 114bpm.mp3	Kick
asdasda	Kick
chord 2 - cruel 67bpm.mp3	Kick
ah - mind 96bpm.mp3	Kick
ah - mind 96bpm.mp3	asdasda

chord 2 - cruel 67bpm.mp3	asdasda
chord 2 - cruel 67bpm.mp3	Closed Hihat
asdasda	Closed Hihat
asdasda	Open Hihat
bass 1- swag 114bpm.mp3	Closed Hihat
chord 2 - cruel 67bpm.mp3	Open Hihat
ah - mind 96bpm.mp3	Fx
asdasda	Fx
bass 1- swag 114bpm.mp3	Open Hihat
ah - mind 96bpm.mp3	Open Hihat
ah - mind 96bpm.mp3	Closed Hihat

Inside Register.txt we can find the following values:

FirstName: Jessie
LastName: J
Username: jessiej
Email: bledea.bogdan97@gmail.com, mark@ronson.com, admin@splicer.com, email, @gmail.com, jessiej@email.com
Password: parola1.

After using the pict cmd we obtain the following table:

FirstName	LastName	Username	Email	Password
Jessie	J	jessiej	mark@ronson.com	parola1.
Jessie	J	jessiej	bledea.bogdan97@gmail.com	parola1.
Jessie	J	jessiej	admin@splicer.com	parola1.
Jessie	J	jessiej	email	parola1.
Jessie	J	jessiej	@gmail.com	parola1.
Jessie	J	jessiej	jessiej@email.com	parola1.

Test Design Techniques - Bugs

I. **Sound is playing many times when matrix cell is clicked** (definitely an implementation bug, we got to clear the sound stream before another sound is playing)
II. **When you close the application and enter again the user is still logged in** (we got to log out all the users when the application is closed)

How to reproduce the bugs:

I. **Sound is playing many times when matrix cell is clicked**
    a. Open the application
    b. Login correctly
    c. Open an existing project
    d. Click on 2 different cells of the matrix
    e. The sound should play one above other
II. **When you close the application and enter again, the user is still logged in**
    a. Open the application
    b. Login correctly
    c. Close the application
    d. Open the application again
    e. The user is still logged in