

Individual Project Report

Software Techniques and Technologies

Demilade Ojurongbe
313777
ELEN7046

June 25, 2012

Abstract

This report presents the design and implementation of a tool for software visualisation and education of unit tests. The tool is used to visually display unit tests enhancing and explaining them while still keeping them in context of the entire system. The focus of the report is on the front end of the system.

Contents

1	Introduction	2
2	Individual Contribution	2
3	Project Design	2
3.1	Description	2
3.2	Approach	2
3.3	Design	3
3.4	Evaluation	4
4	Technologies Utilized	4
4.1	JqTrees / JsTrees	4
4.2	JavaScript	4
4.3	JSON	4
4.4	Others	4
5	Conclusion	5
	References	6

List of Figures

1	Tree View	3
2	Details Panel	3

1 Introduction

The concept of unit tests and unit testing has many benefits, apart from assigning blame and quick error discovery. Unit testing makes code more modular, its easier to use and maintain and it makes development faster [1]. Unit testing is good programming practice and is taught to young developers to enable them write efficient code. This report focuses on an application designed to visualise unit testing.

The designed application is an online system aimed at both developers and student developers who are using or wanting to use unit tests or understand the use of unit tests in an existing system. The application is broken up into back and front end. The back end reflects over the application under test, identifying its unit tests and scenario details, converts this information into JSON data and then passes this to the front end. The front end of the application then breaks down the data, displaying the respective classes, the tests and scenarios and finally a panel of all the information of a scenario including the given, when and then.

The benefits of unit testing to software engineering make this project relevant as it can assist in promoting an important concept in good software engineering practice. This report provides a break down of the application designed and focusses on the development and implementation of the front end of the application.

Section two of this document identifies the individual responsibility, section three explains the design and implementation of the front end, section four identifies the technologies utilised and section five provides the conclusion.

2 Individual Contribution

The designed application is to display information on the unit tests in the application under test. The role this report explains was to iterate through the data which was received in JSON data structure, and displaying it as trees showing the unit tests and scenarios for the selected class, and corresponding the data with the details panel which then displays all data for the selected class. The approach and implementation of these are explained below.

3 Project Design

The focus of this report is on the front end of the system, the web browser. The browser was designed in HTML5 with other developers. As a developer on the team, more focus was placed on the trees and display of the trees.

3.1 Description

The entire project focuses on developing a unit test visualisation application. This application aims to visually represent the classes by reflecting on the application under test as described in the group report. The problem description for this report is then the way in which to display the information on the web page. The data which generated when a class is clicked must be handled to display the corresponding data in the trees. This data must then be sent through to the details panel when the nodes of the tree are clicked, displaying all the data on the clicked or selected class.

3.2 Approach

The tree view panel. Initially empty, will then display the respective tree of the class clicked or highlighted. This shows the user the tree of the class, with the tests for that class and the scenarios for that class. Clicking on another class generates the tree view control for that class and the newly highlighted class's tests and test scenarios are then displayed. The above technologies, JavaScript, JSON and JqTress or JsTrees were used to implement the trees. Figure 1 shows how a tree is displayed for a selected class.

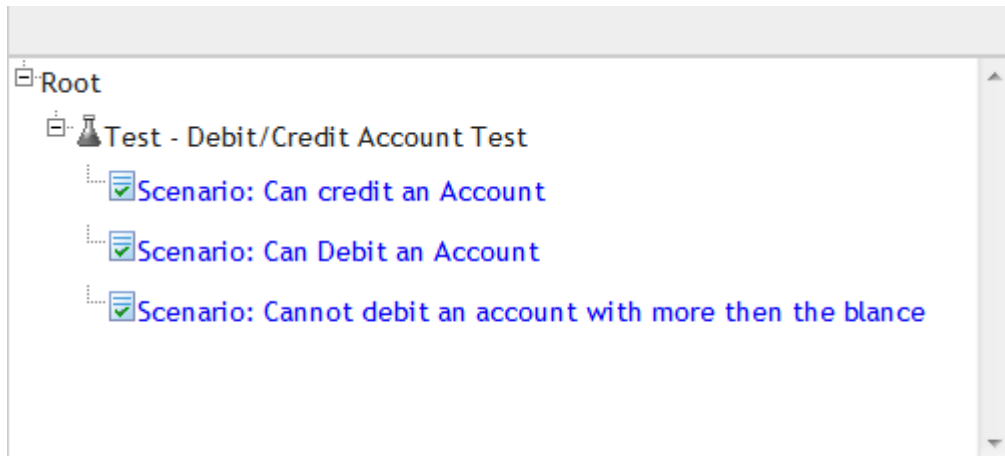


Figure 1: Tree View

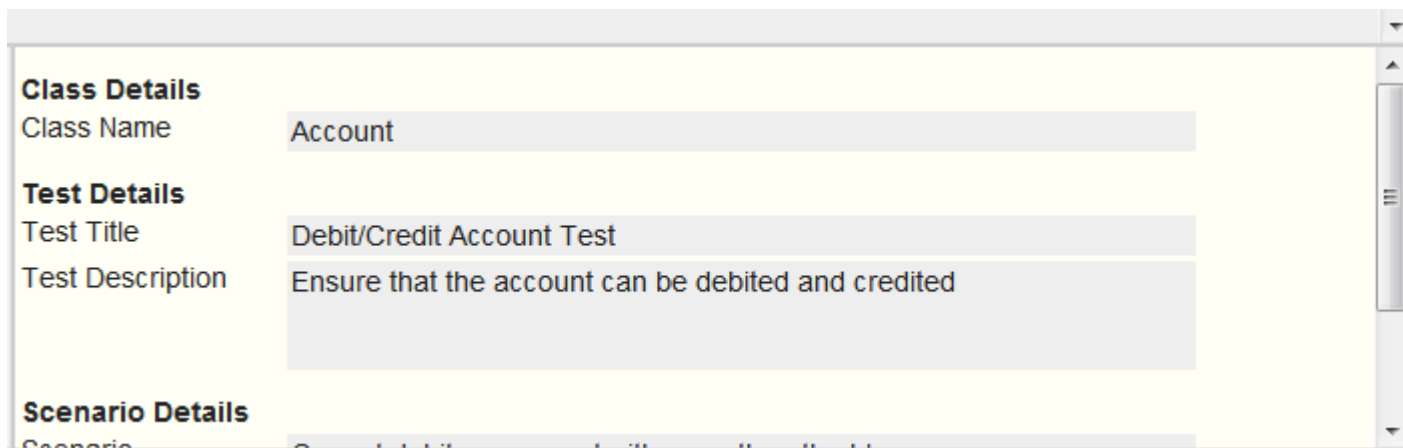


Figure 2: Details Panel

Figure 2 above shows the populated details panel.

To render the trees and the details panel, the approach was further broken down into subtasks.

- Read in data received from the back end of the application.
- Retrieve the Unit tests and scenarios from the data.
- Draw the trees in the allocated panel on the web page.
- Ensure each class has it's tree generated when clicked.
- Add important features for selecting and collapsing the tree.
- Send data to the Details Panel when the tree nodes are clicked.

3.3 Design

Implementation of the above approach.

The client side / web browser is written in HTML5, using the canvass control to create the different panels and handle the different objects to be displayed on the browser screen. The trees are to be displayed on this screen and to do this JavaScript is used. JavaScript as defined previously is a scripting language that can be used to add more functionality to the web page. It also makes the web page / user interface more interactive. The trees are written in JavaScript using the jquery Jstree and JqTree plug in.

JqTrees were initially used for the trees, but a problem was encountered with the tree view control which would kept blinking instead of staying on the screen. The decision was then taken with the lead developer to change the JqTrees to JsTrees to fix this problem.

The tree receives data in JSON data from the back end when a class is clicked. This JSON data contains all the information about the class. The tree view controller, on receiving the data then loops through the data to access the relevant parts of the data which it needs. The tree needs the tests and the scenarios for the tests. The data on the tests and scenarios is extracted and the tree is then built using the data. The tree is then displayed.

The tree has the selectable feature to allow the user to click on it and when this happens the data for that class is sent on to the details panel.

The details panel then displays all the generated and given data for the particular class that is being viewed. The details panel displays the class name, the test name and description, the given , when and then for each scenario under a specific test.

3.4 Evaluation

The design and implementation of the project was done at an appropriate level with possible improvements. During this period, new technologies and methodologies were learnt and used. JavaScript trees, JSON, test driven development, unit testing and HTML5 canvass control were all learnt and used. These new technologies and methodologies were used to improve the design and implementation process and will be used going forward. C sharp was a new programming language used, though initially different was learnt and implemented to a satisfactory degree.

The design and implementation also highlighted important real world scenarios that can add, reduce or completely defeat a teams efficiency. Working with individuals does not mirror working with computers where and expected output is known dependent on the input, people are not as constant and can exceed or deceed expected output.

The project also highlighted the value and importance of adequate and accurate documentation. Excellent designs can be rendered futile with inadequate or barren documentation and vice versa.

4 Technologies Utilized

The technologies and techniques used in the project are mentioned below.

4.1 JqTrees / JsTrees

JqTrees and JsTrees are both JavaScript trees. They have tree view controllers that are used to create the tree from data passed to it and then display the tree. These trees have features like selectable and expandable which make them more interactive. They are packaged as jQuery plugins. This is used in the project to display test and scenario information. They are licensed under the GPL version 2 [2], enabling us to use it for free.

4.2 JavaScript

JavaScript is a scripting language, with functions primarily used on the client side, implemented as part of a web browser to give enhanced user interfaces and dynamic websites [3]. JavaScript enables dynamic use of the web pages, manipulating received data in the application and manipulating it. JavaScript has other uses but its main use is in web pages. It was originally developed in Netscape and is a trademark of the Oracle Corporation, but under licence with Netscape Communications [3].

4.3 JSON

JSON or JavaScript Object Notation is a data interchange format [4]. It is easy for developers to read and write it, it is also easy for machines to parse and generate it [4]. This was the first choice in the data structure to be used, its simplicity and compatibility was a bonus as the application was written by different developers and so data can be easily passed in a single format. JSON is completely language independent and written in text format.

4.4 Others

Other important mentions are:

1. Test Driven Development
2. Peer Programming
3. HTML5

As a group the sections of the back end were done with the lead developer using test driven development to create the tests under a peer programming environment. The work was not always divided into silo's of work but rather done using peer programming to ensure the entire group understood the vital parts of the system and also enhance the learning aspects for the group. A lot was learnt during the peer programming sessions.

HTML5, used for the client side browser. The canvass control element was also used to manipulate the canvass to place the tree in the right section and display it. HTML5 was used to create the details panel, this received data from the tree and displayed it. This needed no additional functionality or interactivity.

5 Conclusion

The document discussed the design and implementation of the front end of the application, placing particular focus on the rendering of tree diagrams with the information from the back end. New technologies were learnt during design and implementation. TDD and Unit testing were vague concepts before the project began, they no longer are. The application has areas of improvement and can be expanded into an even more exciting tool for software visualization and education with great benefit for its users. For time sheet, please refer to group documentation.

References

- [1] The GITS Blog, Various authors (2012), “The benefits of unit testing”, *Random scribbling about programming*, <http://ginstrom.com/scribbles/2007/10/17/the-benefits-of-unit-testing/>, Last accessed June 2012.
- [2] JsTrees, Bozhanov,I.(2012), “JsTrees, jquery tree plugin”, *What is a JsTree?*, <http://www.jstree.com/>, Last accessed June 2012.
- [3] Wikipedia, Various authors (2012), “JavaScript”, *JavaScript*, <http://en.wikipedia.org/wiki/JavaScript>, Last accessed June 2012.
- [4] JSON (2012), “Introducing JSON”, *JavaScript Object Notation*, <http://www.json.org/>, Last accessed June 2012.