Filipe André Cancela Fortuna

JAVA FOR WEB REPORTING

ANEXOS
Web Reporting

Requirement Specification

Web Reporting - SPOTS

Author: Filipe Fortuna
In addition to the authors named on the cover page the following persons have collaborated on this document:

Hugo Magalhães
Manuel Rodrigues
Nuno Alves
Index

0 GENERAL INFORMATION ......................................................................................... 7
  0.1 ISSUE CONTROL............................................................................................. 7
  0.2 HISTORY ........................................................................................................ 7
  0.3 REFERENCES .................................................................................................. 7
  0.4 GLOSSARY AND ABBREVIATIONS ............................................................. 8
  0.5 KEYWORD / DESCRIPTOR ........................................................................... 9
  0.6 LIST OF FIGURES AND TABLES .................................................................. 9

1 INTRODUCTION .................................................................................................... 10
  1.1 SCOPE OF DOCUMENT ................................................................................ 10
  1.2 OBJECTIVES ................................................................................................ 10

2 FEATURES LIST .................................................................................................. 11
  2.1 FUNCTIONAL FEATURES ............................................................................. 11
  2.2 NON-FUNCTIONAL FEATURES ....................................................................... 13

3 FEATURES DESCRIPTION .................................................................................... 15
  3.1 FEA0001 - DEFINING CHART PROPERTIES OPTIONS ................................. 15
      3.1.1 Introduction ......................................................................................... 15
      3.1.2 Requirements .................................................................................... 15
      3.1.3 Description ......................................................................................... 15
      3.1.4 Impacts on other features ..................................................................... 15
      3.1.5 Remarks ............................................................................................. 15
  3.2 FEA0002 - DEFINING CHART PROPERTIES APPEARANCE ......................... 16
      3.2.1 Introduction ......................................................................................... 16
      3.2.2 Requirements .................................................................................... 16
      3.2.3 Description ......................................................................................... 16
      3.2.4 Impacts on other features ..................................................................... 16
      3.2.5 Remarks ............................................................................................. 16
  3.3 FEA0003 - DEFINING TABLE PROPERTIES OPTIONS .................................... 17
      3.3.1 Introduction ......................................................................................... 17
      3.3.2 Requirements .................................................................................... 17
      3.3.3 Description ......................................................................................... 17
      3.3.4 Impacts on other features ..................................................................... 17
      3.3.5 Remarks ............................................................................................. 17
  3.4 FEA0004 - DEFINING TABLE PROPERTIES APPEARANCE ............................. 18
      3.4.1 Introduction ......................................................................................... 18
      3.4.2 Requirements .................................................................................... 18
      3.4.3 Description ......................................................................................... 18
      3.4.4 Impacts on other features ..................................................................... 18
      3.4.5 Remarks ............................................................................................. 18
  3.5 FEA0005 - DEFINING LAYOUT CONFIGURATION ........................................... 19
      3.5.1 Introduction ......................................................................................... 19
      3.5.2 Requirements .................................................................................... 19
      3.5.3 Description ......................................................................................... 19
      3.5.4 Impacts on other features ..................................................................... 19
      3.5.5 Remarks ............................................................................................. 19
  3.6 FEA0006 - ASSOCIATING PRIVATE/PUBLIC TEMPLATE TO A REPORT ........... 20
      3.6.1 Introduction ......................................................................................... 20
      3.6.2 Requirements .................................................................................... 20
      3.6.3 Description ......................................................................................... 20
      3.6.4 Impacts on other features ..................................................................... 20
      3.6.5 Remarks ............................................................................................. 20
  3.7 FEA0007 - MANIPULATING PRIVATE/PUBLIC TEMPLATE ................................ 21
      3.7.1 Introduction ......................................................................................... 21
      3.7.2 Requirements .................................................................................... 21
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7.3</td>
<td>Description</td>
<td>21</td>
</tr>
<tr>
<td>3.7.4</td>
<td>Impacts on other features</td>
<td>21</td>
</tr>
<tr>
<td>3.7.5</td>
<td>Remarks</td>
<td>21</td>
</tr>
<tr>
<td>3.8</td>
<td>FEA0008 - MANIPULATING MULTIPLE REPORTS VIEW</td>
<td>22</td>
</tr>
<tr>
<td>3.8.1</td>
<td>Introduction</td>
<td>22</td>
</tr>
<tr>
<td>3.8.2</td>
<td>Requirements</td>
<td>22</td>
</tr>
<tr>
<td>3.8.3</td>
<td>Description</td>
<td>22</td>
</tr>
<tr>
<td>3.8.4</td>
<td>Impacts on other features</td>
<td>22</td>
</tr>
<tr>
<td>3.8.5</td>
<td>Remarks</td>
<td>22</td>
</tr>
<tr>
<td>3.9</td>
<td>FEA1001 - REPORT VISUALIZATION</td>
<td>23</td>
</tr>
<tr>
<td>3.9.1</td>
<td>Introduction</td>
<td>23</td>
</tr>
<tr>
<td>3.9.2</td>
<td>Requirements</td>
<td>23</td>
</tr>
<tr>
<td>3.9.3</td>
<td>Description</td>
<td>23</td>
</tr>
<tr>
<td>3.9.4</td>
<td>Impacts on other features</td>
<td>23</td>
</tr>
<tr>
<td>3.9.5</td>
<td>Remarks</td>
<td>23</td>
</tr>
<tr>
<td>3.10</td>
<td>FEA1002 - GROUP BY</td>
<td>24</td>
</tr>
<tr>
<td>3.10.1</td>
<td>Introduction</td>
<td>24</td>
</tr>
<tr>
<td>3.10.2</td>
<td>Requirements</td>
<td>24</td>
</tr>
<tr>
<td>3.10.3</td>
<td>Description</td>
<td>24</td>
</tr>
<tr>
<td>3.10.4</td>
<td>Impacts on other features</td>
<td>24</td>
</tr>
<tr>
<td>3.10.5</td>
<td>Remarks</td>
<td>24</td>
</tr>
<tr>
<td>3.11</td>
<td>FEA1003 - SORTING AND FILTERING IN TABULAR REPORTS</td>
<td>25</td>
</tr>
<tr>
<td>3.11.1</td>
<td>Introduction</td>
<td>25</td>
</tr>
<tr>
<td>3.11.2</td>
<td>Requirements</td>
<td>25</td>
</tr>
<tr>
<td>3.11.3</td>
<td>Description</td>
<td>25</td>
</tr>
<tr>
<td>3.11.4</td>
<td>Impacts on other features</td>
<td>25</td>
</tr>
<tr>
<td>3.11.5</td>
<td>Remarks</td>
<td>25</td>
</tr>
<tr>
<td>3.12</td>
<td>FEA1004 - PRINT REPORT</td>
<td>26</td>
</tr>
<tr>
<td>3.12.1</td>
<td>Introduction</td>
<td>26</td>
</tr>
<tr>
<td>3.12.2</td>
<td>Requirements</td>
<td>26</td>
</tr>
<tr>
<td>3.12.3</td>
<td>Description</td>
<td>26</td>
</tr>
<tr>
<td>3.12.4</td>
<td>Impacts on other features</td>
<td>26</td>
</tr>
<tr>
<td>3.12.5</td>
<td>Remarks</td>
<td>26</td>
</tr>
<tr>
<td>3.13</td>
<td>FEA1005 - PDF GENERATION</td>
<td>27</td>
</tr>
<tr>
<td>3.13.1</td>
<td>Introduction</td>
<td>27</td>
</tr>
<tr>
<td>3.13.2</td>
<td>Requirements</td>
<td>27</td>
</tr>
<tr>
<td>3.13.3</td>
<td>Description</td>
<td>27</td>
</tr>
<tr>
<td>3.13.4</td>
<td>Impacts on other features</td>
<td>27</td>
</tr>
<tr>
<td>3.13.5</td>
<td>Remarks</td>
<td>27</td>
</tr>
<tr>
<td>3.14</td>
<td>FEA1006 - DRILL DOWN</td>
<td>28</td>
</tr>
<tr>
<td>3.14.1</td>
<td>Introduction</td>
<td>28</td>
</tr>
<tr>
<td>3.14.2</td>
<td>Requirements</td>
<td>28</td>
</tr>
<tr>
<td>3.14.3</td>
<td>Description</td>
<td>28</td>
</tr>
<tr>
<td>3.14.4</td>
<td>Impacts on other features</td>
<td>28</td>
</tr>
<tr>
<td>3.14.5</td>
<td>Remarks</td>
<td>28</td>
</tr>
<tr>
<td>3.15</td>
<td>FEA1007 - HIDE/SHOW COLUMNS IN TABULAR REPORTS</td>
<td>29</td>
</tr>
<tr>
<td>3.15.1</td>
<td>Introduction</td>
<td>29</td>
</tr>
<tr>
<td>3.15.2</td>
<td>Requirements</td>
<td>29</td>
</tr>
<tr>
<td>3.15.3</td>
<td>Description</td>
<td>29</td>
</tr>
<tr>
<td>3.15.4</td>
<td>Impacts on other features</td>
<td>29</td>
</tr>
<tr>
<td>3.15.5</td>
<td>Remarks</td>
<td>29</td>
</tr>
<tr>
<td>3.16</td>
<td>FEA1008 - CHANGE COLUMN ORDER IN TABULAR REPORTS</td>
<td>30</td>
</tr>
<tr>
<td>3.16.1</td>
<td>Introduction</td>
<td>30</td>
</tr>
<tr>
<td>3.16.2</td>
<td>Requirements</td>
<td>30</td>
</tr>
<tr>
<td>3.16.3</td>
<td>Description</td>
<td>30</td>
</tr>
<tr>
<td>3.16.4</td>
<td>Impacts on other features</td>
<td>30</td>
</tr>
<tr>
<td>3.16.5</td>
<td>Remarks</td>
<td>30</td>
</tr>
<tr>
<td>3.17</td>
<td>FEA2001 - PREDEFINED/CUSTOM REPORTS</td>
<td>31</td>
</tr>
<tr>
<td>3.17.1</td>
<td>Introduction</td>
<td>31</td>
</tr>
<tr>
<td>3.17.2</td>
<td>Requirements</td>
<td>31</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.17.3</td>
<td>Description</td>
<td>31</td>
</tr>
<tr>
<td>3.17.4</td>
<td>Impacts on other features</td>
<td>32</td>
</tr>
<tr>
<td>3.17.5</td>
<td>Remarks</td>
<td>32</td>
</tr>
<tr>
<td>3.18</td>
<td>FEA002 - AD-HOC REPORT</td>
<td>33</td>
</tr>
<tr>
<td>3.18.1</td>
<td>Introduction</td>
<td>33</td>
</tr>
<tr>
<td>3.18.2</td>
<td>Requirements</td>
<td>33</td>
</tr>
<tr>
<td>3.18.3</td>
<td>Description</td>
<td>33</td>
</tr>
<tr>
<td>3.18.4</td>
<td>Impacts on other features</td>
<td>33</td>
</tr>
<tr>
<td>3.18.5</td>
<td>Remarks</td>
<td>33</td>
</tr>
<tr>
<td>3.19</td>
<td>FEA003 - OBJECTS FILTERING</td>
<td>34</td>
</tr>
<tr>
<td>3.19.1</td>
<td>Introduction</td>
<td>34</td>
</tr>
<tr>
<td>3.19.2</td>
<td>Requirements</td>
<td>34</td>
</tr>
<tr>
<td>3.19.3</td>
<td>Description</td>
<td>34</td>
</tr>
<tr>
<td>3.19.4</td>
<td>Impacts on other features</td>
<td>34</td>
</tr>
<tr>
<td>3.19.5</td>
<td>Remarks</td>
<td>34</td>
</tr>
<tr>
<td>3.20</td>
<td>FEA001 - LOGIN</td>
<td>35</td>
</tr>
<tr>
<td>3.20.1</td>
<td>Introduction</td>
<td>35</td>
</tr>
<tr>
<td>3.20.2</td>
<td>Requirements</td>
<td>35</td>
</tr>
<tr>
<td>3.20.3</td>
<td>Description</td>
<td>35</td>
</tr>
<tr>
<td>3.20.4</td>
<td>Impacts on other features</td>
<td>35</td>
</tr>
<tr>
<td>3.20.5</td>
<td>Remarks</td>
<td>35</td>
</tr>
<tr>
<td>3.21</td>
<td>FEA002 - USER INFORMATION MANIPULATION</td>
<td>36</td>
</tr>
<tr>
<td>3.21.1</td>
<td>Introduction</td>
<td>36</td>
</tr>
<tr>
<td>3.21.2</td>
<td>Requirements</td>
<td>36</td>
</tr>
<tr>
<td>3.21.3</td>
<td>Description</td>
<td>36</td>
</tr>
<tr>
<td>3.21.4</td>
<td>Impacts on other features</td>
<td>36</td>
</tr>
<tr>
<td>3.21.5</td>
<td>Remarks</td>
<td>36</td>
</tr>
<tr>
<td>3.22</td>
<td>FEA003 - ADMINISTRATOR INFORMATION MANIPULATION</td>
<td>37</td>
</tr>
<tr>
<td>3.22.1</td>
<td>Introduction</td>
<td>37</td>
</tr>
<tr>
<td>3.22.2</td>
<td>Requirements</td>
<td>37</td>
</tr>
<tr>
<td>3.22.3</td>
<td>Description</td>
<td>37</td>
</tr>
<tr>
<td>3.22.4</td>
<td>Impacts on other features</td>
<td>37</td>
</tr>
<tr>
<td>3.22.5</td>
<td>Remarks</td>
<td>37</td>
</tr>
<tr>
<td>4</td>
<td>NON-FUNCTIONAL FEATURES DESCRIPTION</td>
<td>38</td>
</tr>
<tr>
<td>4.1</td>
<td>FEA001 - QUALITY</td>
<td>38</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Introduction</td>
<td>38</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Requirements</td>
<td>38</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Description</td>
<td>38</td>
</tr>
<tr>
<td>4.1.4</td>
<td>Impacts on other features</td>
<td>38</td>
</tr>
<tr>
<td>4.1.5</td>
<td>Remarks</td>
<td>39</td>
</tr>
<tr>
<td>4.2</td>
<td>FEA002 - SECURITY</td>
<td>39</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Introduction</td>
<td>39</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Requirements</td>
<td>39</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Description</td>
<td>39</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Impacts on other features</td>
<td>39</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Remarks</td>
<td>39</td>
</tr>
<tr>
<td>4.3</td>
<td>FEA003 - SAFETY</td>
<td>40</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Introduction</td>
<td>40</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Requirements</td>
<td>40</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Description</td>
<td>40</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Impacts on other features</td>
<td>40</td>
</tr>
<tr>
<td>4.3.5</td>
<td>Remarks</td>
<td>40</td>
</tr>
<tr>
<td>4.4</td>
<td>FEA004 - MAINTAINABILITY</td>
<td>41</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Introduction</td>
<td>41</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Requirements</td>
<td>41</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Description</td>
<td>41</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Impacts on other features</td>
<td>41</td>
</tr>
<tr>
<td>4.4.5</td>
<td>Remarks</td>
<td>41</td>
</tr>
<tr>
<td>4.5</td>
<td>FEA005 - RESPONSIVENESS</td>
<td>42</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Introduction</td>
<td>42</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Requirements</td>
<td>42</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Description</td>
<td>42</td>
</tr>
<tr>
<td>4.5.4</td>
<td>Impacts on other features</td>
<td>42</td>
</tr>
<tr>
<td>4.5.5</td>
<td>Remarks</td>
<td>42</td>
</tr>
</tbody>
</table>
0 General Information

0.1 Issue Control

The document comprises 58 pages.

0.2 History

<table>
<thead>
<tr>
<th>Issue</th>
<th>Date</th>
<th>Reason for Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12.03.2004</td>
<td>Creation</td>
</tr>
<tr>
<td></td>
<td>25.03.2004</td>
<td>Review</td>
</tr>
</tbody>
</table>

Table 0-1: History

0.3 References

01 NetManager Application Performance Management SPOTS V8.0W


## 0.4 Glossary and Abbreviations

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPOTS</td>
<td>Performance Management application</td>
</tr>
<tr>
<td>Counter</td>
<td>Is a kind of variable that allows the evaluation of the different objects.</td>
</tr>
<tr>
<td>Object</td>
<td>Belongs to a determined class and represents some network component.</td>
</tr>
<tr>
<td>Report</td>
<td>Data retrieved by SPOTS.</td>
</tr>
</tbody>
</table>
0.5 Keyword / Descriptor


0.6 List of Figures and Tables

Figure 5-1: Authentication diagram ................................................................. 49
Figure 5-2: Report presentation template use case diagram .................................. 51
Figure 5-3: Multiple reports view use case diagram ............................................. 53
Figure 5-4: Report visualization use case diagram .............................................. 53
Figure 5-5: Report use case diagram .................................................................. 54
Figure 5-6: Chart properties options sequence diagram ...................................... 57
Figure 5-7: Chart properties appearance sequence diagram ............................... 58
Figure 5-8: Table properties options sequence diagram ..................................... 59
Figure 5-9: Table properties appearance sequence diagram .............................. 60
Figure 5-10: Insert required parameters for custom/predifined reports execution sequence diagram .......................................................... 61
Figure 5-11: Insert network objects sequence diagram ...................................... 62
Figure 5-12: General sequence diagram ............................................................ 62
Figure 5-13: Insert required parameters for ad-hoc reports execution sequence diagram .......................................................... 63

Table 0-1: History ................................................................................................. 7
Table 2-1: Functional features list ....................................................................... 12
Table 2-2: Non-functional features list ............................................................... 13
Table 4-1: Software metric [02] ........................................................................ 39
Table 4-2: Object-oriented metric [02] .............................................................. 39
1 INTRODUCTION

1.1 Scope of document

The document describes the functional and non-functional requirements for the Web Reporting Application based on SPOTS. This application must give more speediness and responsiveness to SPOTS. In other words, this offers a way of presenting, defining and executing reports much lighter, that is to say, it's not necessary to exist such a big effort from the client processing. This application is going to provide some extra functionality, focus in report presentation.

SPOTS appear due to the immense growth and harsh competition that characterizes today's telecommunication business. The operators need to take full advantage of their installed equipment. The principle is:

*Only effective management of the network performance can achieve the balance between optimized usage of the network resources and a good quality of service.*

The network elements have the ability to generate extensive data for performance analysis. As this data can easily add up to several gigabytes, a powerful analysis tool is required to collect, process and analyze the huge volume of information.

The SPOTS tool allows the operator to easily implement the required performance analysis tasks and generate all the necessary information for taking the right decisions [01].

1.2 Objectives

The functional and non-functional requirements here described must provide web access to reporting presentation, definition and execution based on SPOTS.
2 FEATURES LIST

The product can be divided into functional and non-functional requirements regarding the support of Report presentation, visualization, definition and execution.
- Report presentation is related with the definition of a template to configure the reports layout.
- The visualization allows operation manipulation of the report output after execution.
- Report definition allows features specification desired to the report.

These reports allow additional information use, like set(s) of objects, counters and virtual counters.
- Using sets you can define a combination of objects or counters to avoid the action of selecting always the same elements.
- Virtual counters use allows the definition of arithmetic operation between multiple counters.

The requirements are described with more detail in section 3.

2.1 Functional features

The functional features are statements of services that the system should provide, how it should react to particular inputs and how to behave in particular situations.

<table>
<thead>
<tr>
<th>Req.</th>
<th>Name</th>
<th>Prio.</th>
<th>Effort</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Package 1: Report Presentation Template</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0001</td>
<td>Defining chart properties options</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0002</td>
<td>Defining table properties options</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0003</td>
<td>Defining chart properties appearance</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0004</td>
<td>Defining table properties appearance</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0005</td>
<td>Defining layout configuration</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0006</td>
<td>Associating private/public template to a report</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0007</td>
<td>Manipulating private/public template</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Multiple reports view</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0008</td>
<td>Manipulating multiple reports view</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Package 2: Report visualization</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA1001</td>
<td>Report visualization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Report Manipulation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA1002</td>
<td>Group by</td>
<td>2</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>FEA1003</td>
<td>Sorting and filtering in tabular reports</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA1004</td>
<td>Print report</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA1005</td>
<td>PDF generation</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA1006</td>
<td>Drill Down</td>
<td>3</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>FEA1007</td>
<td>Hide/Show columns in tabular reports</td>
<td>3</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>FEA1008</td>
<td>Change column order in tabular reports</td>
<td>3</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td><strong>Package 2: Report Definition</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA2001</td>
<td>Predefined/Custom Reports</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA2002</td>
<td>Ad-hoc report</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Requirement Specification

<table>
<thead>
<tr>
<th>Req.</th>
<th>Name</th>
<th>Prio.</th>
<th>Effort</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEA2003</td>
<td>Objects Filtering</td>
<td></td>
<td>3</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>Package 3: Authentication</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA3001</td>
<td>Login</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>FEA3002</td>
<td>User information manipulation</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>FEA3003</td>
<td>Administrator information manipulation</td>
<td></td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2-1: Functional features list**

- **Prio.** Priority (1 = High, 2 = Medium, 3 = Low)
- **M/O** Mandatory/Optional (M = Mandatory, O = Optional)
2.2 Non-functional features

The non-functional features define the constraints on the services or functions offered by the system.

<table>
<thead>
<tr>
<th>Req.</th>
<th>Name</th>
<th>Prio.</th>
<th>Effort</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEA0101</td>
<td>Performance</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0102</td>
<td>Security</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0103</td>
<td>Safety</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0104</td>
<td>Maintainability</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0105</td>
<td>Responsiveness</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0106</td>
<td>Speed</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0107</td>
<td>Usability</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0108</td>
<td>Robustness</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEA0109</td>
<td>Portability</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 2-2: Non-functional features fist*
3 FEATURES DESCRIPTION

3.1 FEA0001 - Defining chart properties options

3.1.1 Introduction

The Web reporting application must allow the definition of chart options properties.

3.1.2 Requirements

- Select chart type
- Select chart border
- Select legend
- Define thresholds value
- Select chart grid

3.1.3 Description

The chart type can be a column, a line, a pie, an area and a stacked bar. The chart border can be etched, black lined, raised, lowered and empty. Legend has other two items:
- Orientation – Automatic, vertical and horizontal;
- Location – Automatic, south, north, east, west.

The thresholds are values defined by the user that appear as horizontal lines in the chart to improve analysis. The grid improves also chart readability.

3.1.4 Impacts on other features

Chart layout of the feature report visualization.

3.1.5 Remarks

None
3.2 FEA0002 - Defining chart properties appearance

3.2.1 Introduction

The web reporting application must allow setting chart appearance definition.

3.2.2 Requirements

- Select chart series colours;
- Select chart background.

3.2.3 Description

A series is the value of the relationship between objects and counters. The chart series colours allow defining which colour a series must be present in the chart.

The chart background allows defining the chart background colour.

3.2.4 Impacts on other features

Chart layout of the feature report visualization.

3.2.5 Remarks

None
3.3 **FEA0003 - Defining table properties options**

### 3.3.1 Introduction

The web reporting application must allow setting table properties options definition.

### 3.3.2 Requirements

- Select cells font;
- Select header font;
- Select header rotation degrees.

### 3.3.3 Description

As you can see the two first requirements are self-descriptors. The last one allows rotating table headers, so if the header names are too long the header can be rotated and save some space. It can have the following values in degrees:

- 0;
- 90;
- 180;
- 270.

### 3.3.4 Impacts on other features

Table layout of the feature report visualization

### 3.3.5 Remarks

None
3.4 **FEA0004 - Defining table properties appearance**

3.4.1 **Introduction**

The web reporting application must allow change table properties appearance definition.

3.4.2 **Requirements**

- Select table header;
- Select date time cells;
- Select object cells;
- Select counters cells.

3.4.3 **Description**

The table header is the header background colour. The date time, object and counters cells allow cells background colour definition.

3.4.4 **Impacts on other features**

Table layout of the feature report visualization

3.4.5 **Remarks**

None
3.5 FEA0005 – Defining layout configuration

3.5.1 Introduction

This feature enables the user defining where to put the chart or/and table in the report.

3.5.2 Requirements

- Define chart/table position;
- Define header and footer file.

3.5.3 Description

The position must be:

- North;
- South;
- Centre;
- East;
- West.

The header and footer file make possible the inclusion of some html file. This file can include images, text or what the user wants.

3.5.4 Impacts on other features

Report visualization chart/table layout

3.5.5 Remarks

None
3.6  **FEA0006 - Associating private/public template to a report**

3.6.1  **Introduction**

This feature is used to associate some report to a specific template. After this the report visualization depends on the associated template.

3.6.2  **Requirements**

- Private/public template;
- Report.

3.6.3  **Description**

The private template can only be used and manipulated by the owner. The public template can be used by all users.

3.6.4  **Impacts on other features**

Report visualization

3.6.5  **Remarks**

None
3.7 FEA0007 - Manipulating private/public template

3.7.1 Introduction

This feature allows creating, editing and removing private/public template.

3.7.2 Requirements

- Create private/public template;
- Edit private/public template;
- Remove private/public template.

3.7.3 Description

The first two use the requirements FEA0001, FEA0002, FEA0003, FEA0004 and FEA0005. The creation, edition and delection are activities which are only performed by the administrator.

3.7.4 Impacts on other features

Report visualization

3.7.5 Remarks

None
3.8 FEA0008 – Manipulating multiple reports view

3.8.1 Introduction

This feature allows the using of multiple reports in same page, improving the analysis and comparison between different reports.

3.8.2 Requirements

- Define reports position.

3.8.3 Description

The user has the possibility of defining the reports position.

3.8.4 Impacts on other features

Report visualization

3.8.5 Remarks

None
3.9  **FEA1001 - Report visualization**

3.9.1  **Introduction**

This feature provides a way of user report visualization.

3.9.2  **Requirements**

- Associating report with a template.

3.9.3  **Description**

The kind of visualization is defined by the template associated with the report. To see a report the user must associate a template to it.

3.9.4  **Impacts on other features**

None

3.9.5  **Remarks**

None
3.10  **FEA1002 – Group by**

### 3.10.1 Introduction

In the table format, each report result is limited to 50 columns:
- When grouping is not selected, the number of objects is unlimited but it can be selected up to 47 counters;
- When grouping is selected, the maximum number of report results is limited to 5.

What concerns the chart format, the maximum number of data series per report result is 15, where data series means (objects x counters):
- When grouping is not selected, the maximum number of data series (objects x counters) is 15;
- When grouping is selected, the maximum number of report results is limited to 5.

### 3.10.2 Requirements

- Group by variable (counter);
- Group by object.

### 3.10.3 Description

In the table format:
- If the 'Group by variable' clause is selected, unlimited objects and up to 5 counters can be selected;
- If the 'Group by object' clause is selected, up to 5 objects and up to 50 counters can be selected.

In the chart format:
- If the 'Group by variable' is selected, up to 15 objects and 5 counters can be selected in ad-hoc reports. Predefined\Custom reports, typically with more than 5 pre-defined output variables, cannot be presented;
- If the 'Group by object' is selected, up to 5 objects and 15 counters can be selected in ad-hoc reports. Predefined\Custom reports with more than 15 pre-defined output variables cannot be presented.

### 3.10.4 Impacts on other features

None

### 3.10.5 Remarks

None
3.11 **FEA1003** - Sorting and filtering in tabular reports

3.11.1 Introduction

This feature can improve data analysis.

3.11.2 Requirements

- Sorting;
- Filtering.

3.11.3 Description

Filtering provides a way of selecting, defining certain conditions, between table data. With Sorting the user shall be able to sort (ascent and decedent) the table data.

3.11.4 Impacts on other features

None

3.11.5 Remarks

None
3.12 FEA1004 - Print report

3.12.1 Introduction

Giving an adequate report layout to printing.

3.12.2 Requirements

- Print report.

3.12.3 Description

None

3.12.4 Impacts on other features

None

3.12.5 Remarks

None
3.13 FEA1005 - PDF generation

3.13.1 Introduction

Giving a way of PDF report generation.

3.13.2 Requirements

- PDF generation.

3.13.3 Description

None

3.13.4 Impacts on other features

None

3.13.5 Remarks

None
3.14  **FEA1006 - Drill Down**

3.14.1 **Introduction**

This feature must provide a way to drill down from one chart to another.

3.14.2 **Requirements**

- Drill Down.

3.14.3 **Description**

If have a long date interval chart, for instance a week, the user can select a more specific date in this interval, like one day, and a detailed chart will be showed.

3.14.4 **Impacts on other features**

None

3.14.5 **Remarks**

None
3.15 FEA1007 - Hide/Show columns in tabular reports

3.15.1 Introduction

Its purpose is to improve analysis by hiding/showing columns in tabular reports.

3.15.2 Requirements

- Hide/Show columns in tabular reports.

3.15.3 Description

Clicking in the mouse's right button, the user shall be able to precede this action.

3.15.4 Impacts on other features

None

3.15.5 Remarks

None
3.16 FEA1008 – Change column order in tabular reports

3.16.1 Introduction

Its purpose is to improve analysis by changing the column order in tabular reports.

3.16.2 Requirements

- Change column order in tabular reports

3.16.3 Description

By dragging, the user shall be able to precede this action.

3.16.4 Impacts on other features

None

3.16.5 Remarks

None
3.17 **FEA2001 – Predefined\Custom Reports**

### 3.17.1 Introduction

This feature allows the user to define the Predefined\Custom reports properties. To proceed with this, the user must set reports parameters to achieve his purposes. This report's advantage is related to the fact that the user doesn't need to know anything about counters.

### 3.17.2 Requirements

- Define date interval;
- Select data source;
- Select granularity;
- Select network objects;
- Select extra parameters;
- Define general;
- Save task.

### 3.17.3 Description

**Date interval**

The user must select between data interval limits (including now) or the keyword like today, yesterday and last week.

**Data source**

The user must choose between:

- Detail;
- Historical totals;
- Historical SPBH – data focuses on the maximum peak hour for the specified date interval.

**Granularity**

It's the time interval between executions of a certain report.

The user must select one of the following options:

- 5 minute;
- 15 minute;
- 30 minute;
- 1 hour;
- 1 day.

**Network objects**

This requirement includes four steps:

1. Select object class;
2. Select individual object instances and/or set of objects;
3. Set or edit or reset Object filter (FEA2003);
4. Select set of objects result output. The user can choose between:
   - Aggregate – Only one result is presented for the whole set;
- Distinct – When the user pretends to have a result for each object of a set.

Extra parameters
Some reports need additional parameters, for example constant values to proceed specific operations.

Define general
The user must input the desired report name. He must select the template he wants to use or create a new template.

Save task
The user must save the definition of the report for later use.

3.17.4 Impacts on other features

Report visualization

3.17.5 Remarks

None
3.18 **FEA2002** – Ad-hoc report

3.18.1 Introduction

This kind of report is used in situations where no predefined report can be used (like selecting counters that aren’t used in any predefined report). The user is responsible for the definition of the counters he wants to use and extra parameters definition isn’t allowed.

3.18.2 Requirements

- Define date interval;
- Select data source;
- Select granularity;
- Select network objects;
- Select available counters;
- Define general;
- Save task.

3.18.3 Description

All requirements are defined in the last feature, except for counters.

*Counters*

The user must select from the measurements the counters desired.

3.18.4 Impacts on other features

Report visualization

3.18.5 Remarks

None
3.19 FEA2003 – Objects Filtering

3.19.1 Introduction
This must be used to search a specific instance in a class.

3.19.2 Requirements
- Objects filtering.

3.19.3 Description
This provides a way of reducing the object instance choices, making the search easier.

3.19.4 Impacts on other features
None

3.19.5 Remarks
None
3.20 **FEA3001 – Login**

3.20.1 **Introduction**

This must provide a way of authentication and only a register user can pass. In another way, this provides user information, which is used by the system to allow or not, the access to specific functionalities.

3.20.2 **Requirements**

- Input username;
- Input password.

3.20.3 **Description**

The user must input the username and password.

3.20.4 **Impacts on other features**

In FEA3002, because the input information is used in user password and username edit.

In FEA0007 and FEA3002, with the purpose of restricting the ordinary user to do some manager activities, like creating, editing and removing public templates, creating new user, removing user, changing user permissions and editing user information.

3.20.5 **Remarks**

None
3.21 FEA3002 - User information manipulation

3.21.1 Introduction

This feature must provide a set of operation related to user and user management.

3.21.2 Requirements

- Change username;
- Change password;
- Create new user;
- Remove user;
- Change user permissions;
- Edit user information.

3.21.3 Description

The first and second features provide the ordinary user the ability to change his user name and password.
The others requirements are performed by the administrator.

3.21.4 Impacts on other features

None.

3.21.5 Remarks

None.
3.22  **FEA3003** - Administrator information manipulation

3.22.1 Introduction

This feature must provide a set of operation related with administrator management.

3.22.2 Requirements

- Create new administrator;
- Remove administrator;
- Change administrator permissions;
- Edit administrator information.

3.22.3 Description

These requirements are performed by the owner.

3.22.4 Impacts on other features

None.

3.22.5 Remarks

None.
# 4 NON-FUNCTIONAL FEATURES DESCRIPTION

## 4.1 FEA0101 - Quality

### 4.1.1 Introduction

A set of quality measures must be defined to ensure a good maintainability, understanding and error preventing.

### 4.1.2 Requirements

- Quality measures.

### 4.1.3 Description

The quality measures are defined in next table.

<table>
<thead>
<tr>
<th>Software metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fan in/Fan-out</td>
<td>Fan-in is a measure of the number of functions that call some other function (say X). Fan-out is the number of functions which are called by function X. A high value for fan-in means that X is tightly coupled to the rest of the design and changes to X will have extensive knock-on effects. A high value for fan-out suggests that the overall complexity of X may be high because of the complexity of the control logic needed to coordinate the called components.</td>
</tr>
<tr>
<td>Length of code</td>
<td>This is a measure of the size of a program. Generally, the larger the size of the code of a program's components, the more complex and error-prone that component is likely to be.</td>
</tr>
<tr>
<td>Cyclomatic complexity</td>
<td>This is a measure of the control complexity of a program. This control complexity may be related to program understand ability.</td>
</tr>
<tr>
<td>Length of identifiers</td>
<td>This is a measure of the average length of distinct identifiers in a program. The longer the identifiers, the more likely they are to be meaningful and hence the more understandable the program.</td>
</tr>
<tr>
<td>Depth of conditional nesting</td>
<td>This is a measure of the depth of nesting of if-statements in a program. Deeply nested if statements are hard to understand and are potentially error-prone.</td>
</tr>
<tr>
<td>Fog index</td>
<td>This is a measure of the average length of words and sentences in documents. The higher the value for the Fog</td>
</tr>
</tbody>
</table>
Table 4-1: Software metric [02]

<table>
<thead>
<tr>
<th>Object-oriented metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth of inheritance tree</td>
<td>This represents the number of discrete levels in the inheritance tree where sub-classes inherit attributes and operations (methods) from super-classes. The deeper the inheritance tree, the more complex the design as, potentially, many different object classes have to be understood to understand the object classes at the leaves of the tree.</td>
</tr>
<tr>
<td>Method fan-in/fan-out</td>
<td>This is directly related to fan-in and fan-out as described above and means essentially the same thing. However, it may be appropriate to make a distinction between calls from other methods within the object and calls from external methods.</td>
</tr>
<tr>
<td>Weighted methods per class</td>
<td>This is the number of methods included in a class weighted by the complexity of each method. Therefore, a simple method may have a complexity of 1 and a large and complex method a much higher value. The larger the value for this metric, the more complex the object class. Complex objects are more likely to be more difficult to understand. They may not be logically cohesive so cannot be reused effectively as super-classes in an inheritance tree.</td>
</tr>
<tr>
<td>Number of overriding operations</td>
<td>These are the number of operations in a super-class which are over-ridden in a sub-class. A high value for this metric indicates that the super-class used may not be an appropriate parent for the sub-class.</td>
</tr>
</tbody>
</table>

Table 4-2: Object-oriented metric [02]

4.1.4 Impacts on other features

Feature FEA0104.

4.1.5 Remarks

None
4.2 FEA0102 - Security

4.2.1 Introduction

Security is the ability of a system to manage, protect, and distribute sensitive information [96].

4.2.2 Requirements

- Data management security;
- Information Security;
- System security.

4.2.3 Description

- The first requirement must perform data protection from unauthorized (accidental or intentional) modification, destruction, or disclosure [07].
- The second requirement must provide the concepts, techniques, technical measures, and administrative measures used to protect information assets from deliberate or inadvertent unauthorized acquisition, damage, disclosure, manipulation, modification, loss, or use [08].
- The last one must restrict the use of objects to certain users [07].

4.2.4 Impacts on other features

None.

4.2.5 Remarks

The higher the security is, the higher is the user credibility.
4.3  **FEA0103 - Safety**

4.3.1  **Introduction**

Safety is the measure of the absence of unsafe software conditions. The absence of catastrophic consequences to the environment [06].

4.3.2  **Requirements**

- Test software modules.

4.3.3  **Description**

The software must guaranty the data integrity.

4.3.4  **Impacts on other features**

None.

4.3.5  **Remarks**

None.
4.4  FEA0104 - Maintainability

4.4.1  Introduction

Maintainability is the ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment [06].

4.4.2  Requirements

- Maintainability.

4.4.3  Description

The system must use fine-grain and self-contained components.

4.4.4  Impacts on other features

None.

4.4.5  Remarks

None.
4.5 FEA0105 - Responsiveness

4.5.1 Introduction

Responsiveness is the degree to which a software system or component has incorporated the user's requirements [09]. In other way responsiveness is the system ability to inform user about what happens in his actions, how long to wait, frees him to do other things while waiting, manages queued event intelligently, performs housekeeping and low-priority tasks in the background, and makes use of idle time to anticipate (and pre-compute) your likely future requests [10].

4.5.2 Requirements

- Verification and validation of requirements specification;

4.5.3 Description

To increase this feature the user must do the verification and validation of the requirements in earlier stages and last stages. System development must concerns responsiveness issues and a final analysis.

4.5.4 Impacts on other features

None.

4.5.5 Remarks

None.
4.6 FEA0106 - Speed

4.6.1 Introduction

Speed is the rate at which a software system or component performs its functions [09].

4.6.2 Requirements

- Time measurements.

4.6.3 Description

The system response must be measured, to do improvements in case of lack of speed.

4.6.4 Impacts on other features

FEA0105 and FEA0107.

4.6.5 Remarks

This makes a system much more user-friendly, improving the level of satisfaction.
4.7 FEA0107 – Usability

4.7.1 Introduction

Usability is related to the easiness with which a user can learn to operate, to prepare inputs and interpret outputs of a system or component [06].

4.7.2 Requirements

- Provide help;
- Intuitive and descriptive interface.

4.7.3 Description

The system must provide help. The user can learn how to operate it. The intuitive and descriptive interface makes the input and output actions easier to understand. In other words, makes the user memorisation easier to do.

4.7.4 Impacts on other features

FEA0105 and FEA0106.

4.7.5 Remarks

None.
4.8 **FEA0108 – Robustness**

4.8.1 **Introduction**

Robustness is the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environment conditions [06].

4.8.2 **Requirements**

- Error messages and exception handlers.

4.8.3 **Description**

The system must display error messages to the users and the respective information to error correction. The user must have information when important exception or problems occurs and provide information to resolution. This will make the users feel less stressed and they will also understand the problems occurred in a much more easily way.

4.8.4 **Impacts on other features**

FEA0107.

4.8.5 **Remarks**

None.
4.9 **FEA0109 – Portability**

4.9.1 **Introduction**

Portability is related to the easiness with which a system or component can be transferred from one hardware or software environment to another [06].

4.9.2 **Requirements**

- Use portable technologies;
- Use standard conventions.

4.9.3 **Description**

The system's technologies must be as portable as possible.

4.9.4 **Impacts on other features**

None.

4.9.5 **Remarks**

None.
5 Use cases

5.1 Use case diagrams

In this section it will be presented the use cases diagrams which support the requirements described in previous section.

![Authentication Diagram](image)

*Figure 5-1: Authentication diagram*
Figure 5-2: Report presentation template use case diagram
Figure 5-3: Multiple reports view use case diagram

Figure 5-4: Report visualization use case diagram
Figure 5-5: Report use case diagram
5.2 Roles

5.2.1 User

It's the ordinary user, some functionality is not allowed. Still, he can do all report visualization, multiple report view and report execution stuff. This has some limitations in the authentication and report presentation template.

5.2.2 Administrator

It's the user responsible for the management of the application and he can create, edit and remove public templates.

5.2.3 Owner

It's the user responsible for the management of the application administrators.

5.3 Use cases

Some of the use cases are not described because they correspond exactly to the requirements.

5.3.1 Change password

The user must input twice his new password. This reduces the probability of occurring an error. The last step is the submission of the information.

5.3.2 Change username

The user must input twice his new username. This reduces the probability of error occurring. The last step is the submission of the information.

5.3.3 Create new user

This case is performed by the administrator to proceed with the new user's insertion in the system, providing access to the application. The administrator must insert the username, password and the user type (permissions).

5.3.4 Remove user

When a user profile isn't needed anymore the administrator can remove it from the system. A list of all the users is displayed. The administrator can search for the pretended user and then remove it.
5.3.5 Change user permissions

The administrator can switch the user type.

5.3.6 Edit user information

This case is useful when the user forgets his username or password. In the case of forgetting the username the administrator can see it and tell it. In the password case the administrator doesn’t see the password but can change it and give it to the user. After this the user knows the password and all he needs is doing the respective authentication, making the password change immediately.
In the long term this case must be done automatically and by e-mail.

5.3.7 Create new administrator

This case is performed by the owner to proceed with the new administrator’s insertion in the system, providing access to the application. The owner must insert the username, password and the user type (permissions).

5.3.8 Remove administrator

When an administrator profile isn’t needed anymore the owner can remove it from the system. An administrator list is displayed. The owner can search for the pretended administrator and then remove it.

5.3.9 Change administrator permissions

The owner can switch the user type.

5.3.10 Edit administrator information

This case is useful when the administrator forgets his username or password. In the case of forgetting the username the owner can see it and tell it. In the password case the owner doesn’t see the password but can change it and give it to the administrator. After this the administrator knows the password and all he needs is doing the respective authentication, making the password change immediately.
In the long term this case must be done automatically and by e-mail.
5.3.11 Chart properties options

To better understand this case, a sequence diagram is presented below.

![Sequence diagram](image)

*Figure 5.6: Chart properties options sequence diagram*

All of these possibilities are optional, the user only select it if he wants.
5.3.12 Chart properties appearance

In this case, the options have a default value. The users only change it if he wants.

Figure 5-7: Chart properties appearance sequence diagram
5.3.13 Table properties options

Like the previous case, options have default values.

Figure 5-8: Table properties options sequence diagram
5.3.14 Table properties appearance

![Diagram showing the sequence of events for selecting different properties of a table]

Figure 5-9: Table properties appearance sequence diagram

5.3.15 Manipulate tasks

This case is used to execute and editing saved tasks. Only after the execution activity the user can see and operating on the report result.
5.3.16 Insert required parameters for custom/predefined reports execution

The user must define all steps represented in the sequence diagram below. All values of different choices are described in the correspondent feature section. Some custom/predefined reports don’t have extra parameters, so the user can’t define it.

![Sequence diagram](image)

**Figure 5-10: Insert required parameters for custom/predefined reports execution sequence diagram**

The diagrams in next page describe the user extra steps to select network objects and in general.
Figure 5-11: Insert network objects sequence diagram

Figure 5-12: General sequence diagram
5.3.17 Insert required parameters for ad-hoc reports execution

The differences between this sequence diagram and the custom/predefined are the counters selection and the extra parameters. This kind of report doesn’t use extra parameters.

Figure 5-13: Insert required parameters for ad-hoc reports execution sequence diagram
Web Reporting

Functional Specification

Web Reporting - SPOTS
In addition to the authors named on the cover page the following persons have collaborated on this document:

Hugo Magalhães
Manuel Rodrigues
Nuno Alves
TABLE OF CONTENTS

0 GENERAL INFORMATION........................................................................................................ 5
  0.1 ISSUE CONTROL.................................................................................................................. 5
  0.2 HISTORY............................................................................................................................. 5
  0.3 REFERENCES......................................................................................................................... 5
  0.4 GLOSSARY AND ABBREVIATIONS .................................................................................. 6
    0.4.1 Glossary......................................................................................................................... 6
    0.4.2 Abbreviations.................................................................................................................. 7
  0.5 KEYWORD / DESCRIPTOR................................................................................................. 8
  0.6 LIST OF FIGURES AND TABLES.......................................................................................... 8

1 APPLICATION ARCHITECTURE.............................................................................................. 9
  1.1 DEPLOYMENT DIAGRAM .................................................................................................. 9
    1.1.1 Client devices ................................................................................................................ 9
    1.1.2 Web server................................................................................................................... 10
    1.1.3 Database server............................................................................................................ 10
    1.1.4 SPOTS server................................................................................................................ 11
  1.2 COMPONENT DIAGRAM................................................................................................... 12
  1.3 DATA BASE RELATIONAL SCHEMA .............................................................................. 13
    1.3.1 User table ..................................................................................................................... 14
      1.3.1.1 Fields......................................................................................................................... 14
      1.3.1.2 Associations............................................................................................................. 14
    1.3.2 Template table ............................................................................................................. 14
      1.3.2.1 Fields......................................................................................................................... 14
      1.3.2.2 Associations............................................................................................................. 14
    1.3.3 Chart table .................................................................................................................... 14
      1.3.3.1 Fields......................................................................................................................... 14
      1.3.3.2 Associations............................................................................................................. 15
    1.3.4 Legend table .................................................................................................................. 15
      1.3.4.1 Fields......................................................................................................................... 15
      1.3.4.2 Associations............................................................................................................. 15
    1.3.5 Series table .................................................................................................................... 15
      1.3.5.1 Fields......................................................................................................................... 15
      1.3.5.2 Associations............................................................................................................. 15
    1.3.6 Table table .................................................................................................................... 15
      1.3.6.1 Fields......................................................................................................................... 15
      1.3.6.2 Associations............................................................................................................. 16
    1.3.7 Report table ................................................................................................................... 16
      1.3.7.1 Fields......................................................................................................................... 16
      1.3.7.2 Associations............................................................................................................. 16
    1.3.8 Custom table ................................................................................................................. 16
      1.3.8.1 Fields......................................................................................................................... 16
      1.3.8.2 Associations............................................................................................................. 16
    1.3.9 Date interval table ........................................................................................................ 17
      1.3.9.1 Fields......................................................................................................................... 17
      1.3.9.2 Associations............................................................................................................. 17
    1.3.10 Object instance table ................................................................................................ 17
      1.3.10.1 Fields....................................................................................................................... 17
      1.3.10.2 Associations........................................................................................................... 17
    1.3.11 Object set table .......................................................................................................... 17
      1.3.11.1 Fields....................................................................................................................... 17
      1.3.11.2 Associations........................................................................................................... 17
    1.3.12 Extra parameters table .............................................................................................. 18
      1.3.12.1 Fields....................................................................................................................... 18
      1.3.12.2 Associations........................................................................................................... 18
    1.3.13 Ad-hoc table .............................................................................................................. 18
      1.3.13.1 Fields....................................................................................................................... 18
      1.3.13.2 Associations........................................................................................................... 18
    1.3.14 Counter table ............................................................................................................. 18
      1.3.14.1 Fields....................................................................................................................... 18
      1.3.14.2 Associations........................................................................................................... 18
    1.3.15 Virtual counter table ................................................................................................. 18

Functional Specification
1.3.15.1 Fields ......................................................................................... 18
1.3.15.2 Associations ........................................................................... 19
1.3.16 Set of counters table .................................................................... 19
1.3.16.1 Fields ...................................................................................... 19
1.3.16.2 Associations ........................................................................... 19
1.3.17 Report serialization table ............................................................... 19
1.3.17.1 Fields ...................................................................................... 19
1.3.17.2 Associations ........................................................................... 19
1.3.18 Template view table ..................................................................... 19
1.3.18.1 Fields ...................................................................................... 19
1.3.18.2 Associations ........................................................................... 20
1.3.19 Multiple report view table .............................................................. 20
1.3.19.1 Fields ...................................................................................... 20
1.3.19.2 Associations ........................................................................... 20
1.4 CLASS DIAGRAM ............................................................................ 21

2 TECHNOLOGICAL ARCHITECTURE ...................................................... 23
2.1 VIEW ................................................................................................ 23
2.2 CONTROLLER ............................................................................... 23
2.3 MODEL ............................................................................................ 23
2.4 DATABASE ...................................................................................... 23

3 ANNEX ................................................................................................. 24
3.1 TECHNOLOGIC ANALYSIS .............................................................. 24
3.1.1 Introduction ................................................................................ 24
3.1.2 Comparison between JavaServer Faces, Struts and Turbine ........... 27
3.1.3 Conclusions ................................................................................ 29
3.2 VIEWS EXAMPLES .......................................................................... 30
3.2.1 Report presentation template ....................................................... 30
3.2.2 Multiple reports view ................................................................. 39
3.2.3 Report visualization ................................................................. 43
0 GENERAL INFORMATION

0.1 Issue Control

The document comprises 50 pages, all pages have issue 01.

0.2 History

<table>
<thead>
<tr>
<th>Issue</th>
<th>Date</th>
<th>Reason for Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10.04.04</td>
<td>Creation</td>
</tr>
</tbody>
</table>

Table 1: History

0.3 References


/04/ Michael Glögl. Home page. Available at http://www.gloeql.de/17.html

/05/ Alexander Prohorenko and Olexiy Prohorenko. Introduction to JavaServer Faces. Available at: http://www.onjava.com/pub/a/onjava/2004/04/07/jsf.html?page=1


/09/ HSQL Database Engine. Available at: http://sourceforge.net/projects/hsqldb/

/10/ Kito D. Mann. JavaServer Faces 1.0 has been released, but why does it matter?. Available at: http://www.jsfcentral.com/editorial/editorial_1.html

## 0.4 Glossary and Abbreviations

### 0.4.1 Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI components</td>
<td>UI components are configurable, reusable elements that compose the user interfaces of JavaServer Faces applications [02].</td>
</tr>
<tr>
<td>Backing beans</td>
<td>Server-side objects associated with UI components [02].</td>
</tr>
</tbody>
</table>
## 0.4.2 Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>SPOTS</td>
<td>PM application</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
</tbody>
</table>
0.5 Keyword / Descriptor
SPOTS, Performance Management, Performance Monitoring, Mobile Agents

0.6 List of Figures and Tables

Figure 1: Web application architecture ................................................................. 9
Figure 2: Component diagram (the colored associations are a visualization improvement and they haven’t a particular meaning) ............................................................................... 11
Figure 3: Backing bean component constitution .................................................... Error! Bookmark not defined.
Figure 4: Data base relational schema ...................................................................... 13
Figure 5: Class diagram ............................................................................................ 21
Figure 6: Create private template view’s example .................................................... 30
Figure 7: Create private/public template sequence diagram .................................... 31
Figure 8: Chart properties options view’s example ................................................ 31
Figure 9: Chart properties options sequence diagram ........................................... 32
Figure 10: Chart properties appearance view’s example ......................................... 32
Figure 11: Chart properties appearance sequence diagram .................................... 33
Figure 12: Table properties options view’s example .............................................. 33
Figure 13: Table properties options sequence diagram ......................................... 33
Figure 14: Table properties appearance view’s example ......................................... 34
Figure 15: Table properties appearance sequence diagram .................................... 34
Figure 16: Associate report to private/public template view’s example .................... 35
Figure 17: Associate report to private/public template sequence diagram ................ 35
Figure 18: View/edit private/public template view’s example .................................. 36
Figure 19: View/edit private/public template sequence diagram .............................. 37
Figure 20: Report presentation template class diagram .......................................... 38
Figure 21: Create multiple reports view view’s example ......................................... 39
Figure 22: Choosing reports view’s example ........................................................ 39
Figure 23: Create multiple reports view sequence diagram .................................... 40
Figure 24: View/edit multiple reports view list view’s example ............................... 41
Figure 25: View/edit multiple reports list sequence diagram .................................. 41
Figure 26: Multiple report view .............................................................................. 42
Figure 27: Report visualization view’s example ..................................................... 43
Figure 28: Report association template options view’s example ............................. 44
Figure 29: Report visualization specifying a filter view’s example .......................... 44
Figure 30: Report visualization filter result view’s example .................................... 45
Figure 31: Multiple reports view visualization view’s example ............................... 46
Figure 32: Multiple reports view visualization specifying a filter view’s example .......... 47
Figure 33: Report visualization class diagram ....................................................... 48
Figure 34: Report visualization sequence diagram ................................................ 49
Figure 35: Report visualization specifying filters sequence diagram ...................... 50

Table 1: History .................................................................................................... 5
Table 2: Technologic table comparison .................................................................. 27
1 APPLICATION ARCHITECTURE

1.1 Deployment diagram
The following picture shows all web application architecture and integration with SPOTS.

![Deployment diagram](image)

Figure 1: Web application architecture [11]

This multi-tiered architecture enables the distribution of application processing effort and increases speed. It also provides more modularity and maintainability.

- Modularity because each node (application device) has a specific function and specific components.
- Maintainability is related with the modularity because well known tier function and components make easy to find where to precede modifications. For other way, modification in a component or tie doesn’t lead to all system changes;

In the next items the different nodes functions are explain.

1.1.1 Client devices
In this node we have the browser that provides a way of using the web application. The browser is responsible for the communication with the server, requesting and providing information to the server. The browser is the user framework, providing all web application tools and allows presenting information to the user.

1.1.2 Web server
This node is based in a Model 2/ Model-View-Controller architecture.

- The view is the output, the user interface and it uses UI Components;
The controller is made of a set of backing beans that perform navigation, event handling, form validation and error handling;

The navigation provides a way knowing web application next view;

Event handling allows the application to know when an event occurs and what to do with it;

Form validation allows validation of the user to input;

Error handling allows the treatment of validation errors.

The model provides the business logic objects. These objects are responsible for the interaction between enterprise java beans, java beans, data and xml.

1.1.3 Data base server

This node provides and receives data to and from the application. It's the persistent tier of the web application.

1.1.4 SPOTS server

This node is responsible for delivering report data in xml files. These data are used to make charts and tables in the report presentation.
1.2 Component diagram

The component diagram below shows the application structure.

Figure 2: Component diagram (the colored associations are a visualization improvement and they haven't a particular meaning)
The diagram is divided in three packages that correspond to three tiers. First I will describe the three-tier component relationship and next I will describe the tier relationship.

- The view is composed by components that display the web pages. The first component the user will see is the login. He must provide his username and password to enter in the application. After this, he has access to an index component, which is a welcome page with a menu. Through it, the user has access to different web application functionalities.

- The middle tier has two packages, the model and the controller.

- The controller is made by backing beans, which provide a way of interaction and manipulation between view and model. This package has six components.
  - The **User backing bean** must interact with the user bean to perform authentication and the user information change or view.
  - The **Multiple report view backing bean** must interact with multiple report view bean to create, edit and remove some of this beans and it also interact with the report visualization bean to get the available reports list.
  - The **Template backing bean** must interact with the template bean with the purpose of creating, editing and removing templates and it also must interact with the report visualization bean to get the available report’s list.
  - The **Report backing bean** must interact with report bean to create, edit and remove report beans.
  - The **Report visualization backing bean** must interact with the report visualization bean to get every properties and features of the selected report. It must interact with the report bean to get the template associated with the wanted report and it must also interact with the template bean to get the template properties.
  - The **Menu backing bean** must interact with the report visualization bean and with the multiple report view bean to get the available reports and the multiple report views list.
  - The model is made of beans and provides the objects wanted by the backing beans to perform operations.

- The data tier provides all the persistent data to the application. It is made of two packages. One of them keeps xml documents provided by SPOTS application and the other are the tables of the data base. The relationship between the different tables in the data base is described in the next section.

In what concerns the tier relationship, the views components receive user requests and input data and give this information to backing beans. Backing beans are responsible for event and listener model that defines how to handle component events, for conversion model that defines how to plug in data converters onto a component and for validation model that defines how to register validators onto a component.

In what concerns the middle tier and the data tier relationship, only data is exchanged between them.
1.3 Data base relational schema

Figure 3: Data base relational schema
1.3.1 User table

Must provide all the information about the application users

1.3.1.1 Fields

This table has seven fields. The first one is the identifier which is a primary key and it must be unique and not null. The other ones are the self-explanatory and all of them are required. The permission field must take the values user, administrator or owner.

1.3.1.2 Associations

One User can have none or more Templates.

1.3.2 Template table

Must provide all the information about Template features

1.3.2.1 Fields

This table has five fields. All of them are required. The foreign key must not be null. The type field must take public or private values. If a Template is private only the creator can use it.

1.3.2.2 Associations

A Template must have between one to five visual representations (Chats and/or Tables). These ones can be mixed too. One Template can be associated with none or many Reports. A template must belong to a single User.

1.3.3 Chart table

Must provide all the information about Chart properties and options

1.3.3.1 Fields

All the fields are required except border, grid and the threshold which can be null. Chart table have a foreign key which must be filled with the associated Template id.

The position field must take the following values:

- North;
- Center;
- South;
- East;
- West.
1.3.3.2 Associations

A Chart can have none or one legend and one or several Series. A Chart must be associated with a Template.

1.3.4 Legend table

Must provide all the information about Legend's properties and options

1.3.4.1 Fields

All the fields are required. Legend table have a foreign key which must be filled with the associated Chart id.

1.3.4.2 Associations

A Legend must be associated with a Chart.

1.3.5 Series table

Must provide all the information about Series properties

1.3.5.1 Fields

All the fields are required. Series table have a foreign key which must be filled with the associated Chart id.

1.3.5.2 Associations

A Series must be associated with a Chart.

1.3.6 Table table

Must provide all the information about table's properties and options

1.3.6.1 Fields

All the fields are required. Table table have a foreign key which must be filled with the associated template id.

The position field must take the following values:
- North;
- Center;
- South;
- East;
- West.
1.3.6.2 Associations

A table must be associated with a template.

1.3.7 Report table

Must provide all the information about report properties

1.3.7.1 Fields

All the fields are required. Report table has a foreign key which must be filled with the associated Template id.

The type field must take normal or periodic values. A report is normal when is generated once, in a specific date and time. It’s periodic when the same report is generated several times according to a specific periodicity (ex: hourly).

1.3.7.2 Associations

A Report must be associated with a Template. A Report can have none or many TemplateView and ReportSerialization.

1.3.8 Custom table

Must provide all the information needed to Custom reports.

1.3.8.1 Fields

All the fields are required. Custom table have a foreign key which must be filled with the associated Report id. The dataSource field must have the detail or historical totals or the historical SPBH values.

The granularity must take the following values:
- 5 minutes;
- 15 minutes;
- 30 minutes;
- 1 hour;
- 1 day.

1.3.8.2 Associations

A Custom must be associated with a Report. One Custom must have the Date interval and one or many Object instance or Object Set. These ones can be mixed too. A Custom can have an Ad-hoc and none or many Extra parameters.

1.3.9 Date interval table

Must provide all the information about the date interval boundaries or keywords
1.3.9.1 Fields
Date interval table has a foreign key which must be filled with the associated Custom id.
The begin and end fields can have interval date boundaries.
The keyword can take the following values:
- Today;
- Yesterday;
- Last week.
Begin and end fields must be specified if keyword field doesn’t and vice-versa.

1.3.9.2 Associations
A Date interval must be associated with a Custom.

1.3.10 Object instance table
Must provide all the information about object’s instance properties

1.3.10.1 Fields
All fields are required. Object instance table have a foreign key which must be filled with the associated Custom id.

1.3.10.2 Associations
An Object instance must belong to a Custom.

1.3.11 Object set table
Must provide all the information about Object’s set properties

1.3.11.1 Fields
All the fields are required. Object set table have a foreign key which must be filled with the associated Custom id.

1.3.11.2 Associations
An Object Set must be associated with a Custom.

1.3.12 Extra parameters table
Must provide all the information about extra parameters properties
1.3.12.1 Fields

All the fields are required. Extra parameters table have a foreign key which must be filled with the associated Custom id.

1.3.12.2 Associations

An Extra parameter must be associated with a Custom.

1.3.13 Ad-hoc table

Must provide all the information about Ad-hoc properties

1.3.13.1 Fields

All the fields are required. Ad-hoc table have a foreign key which must be filled with the associated Custom id.

1.3.13.2 Associations

An Ad-hoc must be associated with a Custom. An Ad-hoc must have one or many Counters or Virtual counters or sets of counters. These ones can be mixed too.

1.3.14 Counter table

Must provide all the information about Counter properties

1.3.14.1 Fields

All the fields are required. Counter table have a foreign key which must be filled with the associated Ad-hoc id.

The measurement field is the measurement where the counter belongs to.

1.3.14.2 Associations

A Counter must belong to an Ad-hoc.

1.3.15 Virtual counter table

Must provide all the information about Virtual counter's properties

1.3.15.1 Fields

All the fields are required. Virtual counter table have a foreign key which must be filled with the associated Ad-hoc id.
1.3.15.2 Associations

A Virtual counter must be associated with an Ad-hoc.

1.3.16 Set of counters table

Must provide all the information about the Set of counters properties

1.3.16.1 Fields

All the fields are required. Set of counters table have a foreign key which must be filled with the associated Ad-hoc id.

1.3.16.2 Associations

A Set of counters must be associated with an Ad-hoc.

1.3.17 Report serialization table

Must keep Report object serialization. These reports objects keep xml reports documents information generated by SPOTS application.

1.3.17.1 Fields

All the fields are required. Report serialization table have two foreign keys which must be filled with the associated Template and Report id.

The objectSerialization field must be used to put a Report object serialization associated to a Template.

1.3.17.2 Associations

A Report's serialization must be associated to one Report.

1.3.18 Template view table

This the table allows the relationship between the Report and the multiple report view.

1.3.18.1 Fields

All the fields are required. Template view table have two foreign keys which must be filled with the associated Report and with the Multiple report view id.

The position field must take the following values:

- North;
- Center;
- South;
- East;
• West.

1.3.18.2 Associations

A template view must be associated with a Report and with a Multiple report’s view.

1.3.19 Multiple report view table

This table keeps the information related with the use case which the user wants to see multiple reports.

1.3.19.1 Fields

All the fields are required.

1.3.19.2 Associations

A Multiple report must be associated between 2 and 5 Template views.
1.4 Class diagram

Figure 4: Class diagram
The model is made of:
- Multiple report;
- Parser XML;
- Report Item;
- Report visualization;
- Counter;
- Object instance;
- Template;
- Chart properties;
- Table properties;
- Filter;
- Report bean.

The model tier function is support controller, interacting and separating controller from data. This provides more modularity and maintainability because the changes in the data tier or in the controller only have effect in the model's tier.

The classes Template, Multiple report and Report visualization backing bean provide a bridge between the model and the view's tier, any changes that can possibly occur in the model don't affect the view.

The class Report visualization bean uses Parser XML Bean objects to transform the xml data into objects. This data is used to construct objects from Counter and Object Instance bean classes. The Object instance bean class must have one or many counters. The Report visualization bean class is sometimes used only to get the available reports list. This action involves Parser XML Bean and Report Item Bean. Almost all classes which are associated with Report visualization bean make use of it because of the above-mentioned reason with the exception of the Report visualization backing bean.

The Template's backing bean has some attributes associated with the view's components to get user choices. These choices are put into Chart or Table properties beans. There can only be a maximum of five objects which can be the Chart or the Table's properties beans or they can be mixed and associated with a Template's backing bean.

All these input data is placed in a Template's bean which is responsible for putting it in the data base. This class uses the User's bean class to get the user permission because only the administrators and the owners can create, edit and remove public Templates.

The Report's visualization backing bean uses the Report's visualization bean to get the object's instances and the counters needed to construct the view. Another source which is used is the Report bean and the Template's bean. The first one retrieves the report identifier and the object's serialization if exists. The second one retrieves template's properties associated with the report.

When the user wants to view a multiple report's view the Report visualization backing bean uses the Multiple report view bean to get the reports specified in the view and get through the Report visualization and Template beans the correspondent data and template properties.

The Multiple report backing bean uses the multiple report bean to put and to get data.

More detailed information about this subject is available in the annex where some views examples, associated sequence and class's diagrams are presented.
2 TECHNOLOGICAL ARCHITECTURE

In this section are described the technologies used in the different tiers. For additional information about this technologies go see Technological analysis annex.

2.1 View

In the view are used Java Server Faces components, html, Java Script, tiles and applets.

2.2 Controller

The controller uses java backing beans which are closed linked with the view tier.

2.3 Model

This tier uses Hibernate which is based in Object-relational mapper (ORM) and helps to bridge the gap between objects and relational data base (object-relational gap) [04]. This technology uses java beans, Hibernate Query Language and XML for mapping.

2.4 Database

In order to develop the database it will be used the Hypersonic SQL data base engine. In later versions will be used the Oracle data base engine.
3 ANNEX

3.1 Technologic analysis

3.1.1 Introduction

This section has the purpose of making a introduction to all technologies planning to be used in WEB application construction. In next section a comparison is made between some of these technologies. This analysis isn't an absolute truth and doesn't include all WEB information. It's the possible analysis with closed time boundaries.

All this technologies implements the MCV (Model-Controller-View).

MCV architecture introduces a controller servlet between the browser and the JSP pages or servlet content being delivered. The controller centralizes the logic for dispatching requests to the next view based on the request URL, input parameters, and application state. The controller also handles view selection, which decouples JSP pages and servlets from one another. MCV applications are easier to maintain and extend, because views do not refer to each other directly. The MCV controller servlet provides a single point of control for security and logging, and often encapsulates incoming data into a form usable by the back-end MVC model. For these reasons, the MCV architecture is recommended for most interactive applications.

JavaServer Faces Technology is a user interface framework for building Web applications. The main components of JavaServer Faces technology are:

- A graphical user interface (GUI) component framework;
- A flexible model for rendering components in different kinds of HTML, or different markup languages and technologies. A Render generates the markup to render the component and converts the data stored in a model object to types that can be represented in a View [02].
- A standard RenderKit for generating HTML/4.01 markup.

The available framework components are:

- UIC components encloses all of the controls that display or collect data from the user;
- UIData component supports data binding to a collection of data objects. The Table’s render displays the data as an HTML table;
- UIColumn component represents a column of data in a UIData component;
- UICommand component performs an action when it is activated;
- UIGraph component displays an image;
- UInput component displays a value to a user and it allows the user to modify this data. The most common example is a text field;
- UOutput component displays data that cannot be modified;
- UIPanel is rendered as an HTML table. This component differs from UIData in that UIData can dynamically add or delete rows to accommodate the underlying data source, whereas a UIPanel must have the number of rows predetermined;
- UISelectBoolean class defines components that have a boolean value;
- The **UISelectMany** class defines components that allow the user to select zero or more values from a set of values;
- **UIMessage** and **UIMessages** components are used to display error messages;
- **UISelectOne** class defines components that allow the user to select one value from a set of values;
- **UISelectItem** and **UISelectItems** represent components that can be nested inside a **UISelectOne** or a **UISelectMany** component. The **UISelectItem** is associated with a SelectItem instance, which contains the value, label, and description of a single item in the **UISelectOne** or **UISelectMany** component. The **UISelectItems** instance represents a set of SelectItem instances, containing the values, labels, and descriptions of the entire list of items.

In support of the GUI components are the following features:

- Managed model object creation;
- Input validation;
- Event handling;
- Data conversion between model objects and components;
- Page navigation configuration.

The core JSF architecture is designed to be independent of communication protocols or markup language specifics. However, it's also meant to solve the problems experienced working with HTML clients communicating via HTTP with a Java application server that supports Servlet/JSP applications [05].

The **Struts** framework is a flexible control layer based on standard technologies like Java Servlets, JavaBeans, ResourceBundles, and XML, as well as various Jakarta Commons packages. Struts encourage application architectures based on the MVC approach, a variation of the classic Model-View-Controller (MVC) design paradigm.

Struts provide its own Controller component and integrate with other technologies to provide the Model and the View. For the Model, Struts can interact with standard data access technologies, like JDBC and EJB, as well as most any third-party packages, like Hibernate, iBATIS, or Object Relational Bridge. For the View, Struts works well with JavaServer Pages, including JSTL and JSF, as well as Velocity Templates, XSLT, and other presentation systems.

**Turbine** is a servlet based framework that is excellent for developing applications that make use of a services-oriented architecture. Some of the functionality provided with Turbine includes a security management system, a scheduling service, XML-defined form validation server, and an XML-RPC service for web services. Parts of Turbine can also be used independently of the web portion of Turbine as well. This situation allows the use in others applications.

The Turbine core is free of any dependency on a presentation layer technology. Both JavaServer Pages (JSP) and Velocity are supported inside Turbine.

**Hibernate** is an object-relational mapper (ORM), which help bridge the gap between object-orientated programming and relational databases. The relational model deals with relations, tuples and sets. The object-orientated paradigm however deals with objects, their attributes and associations to each other. Hibernate makes objects persistent using a relational database.

Typically to make an object persistent the developer must open a JDBC connection, create an SQL statement and copy all your property values over to the select. This may be a hard work.

Some studies showed that 35% of an application code was produced by the mapping between application data and the data store [04].
Hypersonic SQL Data base is simple but very fast data base engine written in Java, with a JDBC driver, supporting a subset of ANSI-92 SQL. It offers a small (about 100k), database engine which offers both in memory and disk based tables. This product includes Hypersonic SQL.
3.1.2 Comparison between JavaServer Faces, Struts and Turbine

<table>
<thead>
<tr>
<th>Web applications framework</th>
<th>Maintainability</th>
<th>Robustness</th>
<th>Portability</th>
<th>Easy learning</th>
<th>Competitive Advantage</th>
<th>Available Documentation</th>
<th>Easy implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSF</td>
<td>*****</td>
<td>***</td>
<td>****</td>
<td>*****</td>
<td>*****</td>
<td>*****</td>
<td>*****</td>
</tr>
<tr>
<td>Struts</td>
<td>****</td>
<td>*****</td>
<td>*****</td>
<td>***</td>
<td>***</td>
<td>*****</td>
<td>***</td>
</tr>
<tr>
<td>Turbine</td>
<td>****</td>
<td>****</td>
<td>*****</td>
<td>****</td>
<td>**</td>
<td>*****</td>
<td>*****</td>
</tr>
</tbody>
</table>

Table 2: Technologic table comparison.

- **Maintainability**
  - All technologies are based in Model View Controller (MVC) architecture that allows the developer to separate presentation from business logic, which makes more maintainable code. JSF is better than others because uses UI components.

- **Robustness**
  - Struts and Turbine have four years of consecutive development and using.
  - JSF had two years of development, but was made by the main responsible of Struts implementing and have all experience provided by this situation.

- **Portability**
  - All technologies are JAVA (Servlets/JSP) based.

- **Easy learning**
  - The JSF allows more abstraction and simplify the MCV model.
Competitive advantage

The JSF is a newer technology that has in perspective a good future. In other way, JSF had been made by the main struts author, implementing in JSF a struts philosophy improvement and correct all lacks and errors. JSF has all struts potentialities and some extra capabilities. Proponents say JSF is a huge leg up with Web application development, from hard-coding markup-languages like XML and HTML, and provides more flexibility than Struts, which can only render an element one way [01]. More importantly, JSF is an attempt to fill a hole in the standard Java stack – a hole that .NET has had fully fleshed out since day one [10].

Available Documentation

The JFS has less documentation because it's the earlier. But Sun tutorial [02] is very complete and fulfills all needs of understanding JSF features and capabilities.

Easy implementation

One of the main and important differences between JSF and Struts is the struts MCV implementation simplification.

Turbine is similar to struts.

3.1.3 Conclusions

In technologic business we have to take attention to the incoming technologies, all his capabilities and improvements.

As we can see in the technologies analysis, JSF improved and reduced struts complexity. In another way, it includes some .NET features to fulfill JAVA web frameworks lacks.

The GUI in web reporting application is very important. The better framework in this skill is JSF.

For the above reasons the chosen framework is JSF.
3.2 Views examples

3.2.1 Report presentation template

When the mouse pointer is over the report's presentation template menu button, three options are displayed:

- Create private template;
- Associate report to private/public template;
- View/edit private/public template list.

Selecting the first option the user has access to the view below. The user can select how to display the report data (chart or table), the respective position and if he wants to include some file in the header and/or the footer, to personalize the report's visualization. By clicking in the "properties" button the user can change the properties and options which are associated with the chart or table, depending on the selected item in the combo box. By pressing the "create" button the user can create the template with the name he pretends.

![Web Reporting](image)

Figure 5: Create private template view's example
The figure below is an example of a chart’s properties specification. The chart’s name and type have default value and the other properties are optional. The border’s option default value is none, the user can change it if he wants to use it. The other ones can be used checking the respective check box. In the legend and thresholds the options Orientation, Location and value are only available when the checking has been done.

By clicking in the “next” button the user has access to the chart’s properties appearance view which is presented in the figure 10.

![Chart properties options](image)

Figure 7: Chart properties options view's example
In this step the user can define the color to the background and the series. These options have a default values.

By clicking in the "Ok" button the user finishes the properties editing and he returns to the figure 6. Once there, he can edit more properties or finish this process by pressing create button.

**Web Reporting**

**Chart properties appearance**

```
<table>
<thead>
<tr>
<th>Report Presentation Template</th>
<th>background: Green ▼</th>
</tr>
</thead>
<tbody>
<tr>
<td>Series:</td>
<td></td>
</tr>
<tr>
<td>Serie 1</td>
<td>Green ▼</td>
</tr>
<tr>
<td>Serie 2:</td>
<td>Green ▼</td>
</tr>
<tr>
<td>Ok</td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 9: Chart properties appearance view's example
Below there is an example of the table’s properties edition. Where the default values can be changed to the different options. By clicking “next” button the user has access to the table's properties appearance view.

**Web Reporting**

**Table properties options**

<table>
<thead>
<tr>
<th>Report Presentation Template</th>
<th>Table Name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cells font/size:</td>
<td>Table 1</td>
</tr>
<tr>
<td>Times New Roman</td>
<td>12</td>
</tr>
<tr>
<td>Header font/size:</td>
<td>Times New Roman</td>
</tr>
<tr>
<td>Header rotation degrees</td>
<td>0</td>
</tr>
</tbody>
</table>

*next*

**Figure 11:** Table properties options view's example

**Figure 12:** Table properties options sequence diagram
The user can select the color for the different kind of cells. By pressing "Ok" he passes to the create private template view (figure 6).

**Web Reporting**

**Table properties appearance**

<table>
<thead>
<tr>
<th>Report Presentation Template</th>
<th>Table header:</th>
<th>Green</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date time cells:</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>Object cells</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>Counters cells</td>
<td>Green</td>
<td></td>
</tr>
</tbody>
</table>

**Ok**

**Figure 13: Table properties appearance view's example**

**Figure 14: Table properties appearance sequence diagram**
When the user selects in the menu, the report’s presentation template and the sub-item Associate report to private/public template he has access to the view below. The view displays all the available reports with their associated templates. Some reports don’t have any kind of association yet and the user can define them by clicking in the combo box where all the available templates are displayed. The reports which have already a template associated can be changed.

**Web Reporting**

Associate report to private/public template

```
Report 1       None       Ok
Report 2       Template 1  Ok
Report 3       Template 2  Ok
Report 4       Template 1  Ok
```

Figure 15: Associate report to private/public template view’s example.

![Sequence Diagram](image)

Figure 16: Associate report to private/public template sequence diagram.
When the user selects in menu options, the report's presentation template and the sub-item View/edit private/public template list he has access to the view below.

A list of templates is presented. In the case of being a regular user this list have public and private templates but the user only can edit and remove the private. If the user has administrators or owners permissions, can change public and private templates.

By presses the template name the user has access to every properties of the selected template.

By clicks in the "edit" button the user has access to a page similar to the one in figure 6, but with all the properties of the selected template, then he can change it.

By pressing the "remove" button the respective template is removed from the database.

**Web Reporting**

*View/edit private/public template list*

<table>
<thead>
<tr>
<th>Template 1</th>
<th>Edit</th>
<th>Remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template 2</td>
<td>Edit</td>
<td>Remove</td>
</tr>
<tr>
<td>Template 3</td>
<td>Edit</td>
<td>Remove</td>
</tr>
<tr>
<td>Template 4</td>
<td>Edit</td>
<td>Remove</td>
</tr>
</tbody>
</table>

Figure 17: View/edit private/public template view's example.
Figure 18: View/edit private/public template sequence diagram
Figure 18: View/edit private/public template sequence diagram
Almost all the Template's backing bean attributes are associated with UI Components which compound the view, like the name which is associated with a UI input in the figure 6 to get the template name.

Most of these attributes have a default value which is attributed when the user enters in the page and when he clicks in the properties button (figure 6). This default values depends on the item selected by user in the combo box (chart or table). If the user's choices are different from the default, values are keeping by this attributes. The attribute user allows WEB application gets user permission to provide or not the possibility of public template creation.

When the user presses "ok" button in figure 10 or 14, it is created a new chart bean or table bean (createChart(), createTable()), with the values saved by the different attributes. These objects are putted in the ArrayList chartTable.

When the user clicks in the create button (figure 6), it is used the createTemplate() method and with the help of a template's attribute, a new template are putted in the persistent data base.

In the figure where are represented the association between a report and a template (figure 16) the following methods are used:

The getAllReports() returns an array list with the available reports, using the report attribute of the class report visualization bean.

The getAllTemplates() returns an array list with the existing templates in the data base, using the template attribute.

The setReportToTemplate allows establishing an association between a report and a template.

In the figure which corresponds to the view/edit private/public template (figure 18). It is used the method getAllReports. Beyond this method, it's also used the editReport and
the removeTemplate methods. These methods use the template’s variable to update and remove templates.

3.2.2 Multiple reports view

When the mouse pointer is over the multiple report’s view menu button, three options are displayed:

- Create multiple report view;
- View/edit multiple report list.

Selecting the first option the user has access to the view below. Once here he must input a name and between two to five different reports. To choose the reports the user must press the “choose Report” button which provides the access to all the available reports. By clicking in the “create” button he creates a new multiple reports view.

Web Reporting

Figure 20: Create multiple reports view view's example.

The once reports are scheduled for one specified date and hour and are created once. The others have a periodicity, for example the report 1 is created hourly and report 2 daily.

Figure 21: Choosing reports view’s example.
Figure 22: Create multiple reports view sequence diagram.
Selecting the second choice the user has access to the view below. Where it is presented a list of all the multiple reports views available and he can edit, remove and see its properties (pressing the multiple reports view name).

**Web Reporting**

*View/edit multiple reports list*

**Welcome User**
- User information
- Report Presentation Template
- Multiple Reports view
- Report visualization
- Report Execution

Multiple report view 1
- Edit
- Remove

Multiple report view 2
- Edit
- Remove

Multiple report view 3
- Edit
- Remove

Figure 23: View/edit multiple reports view list view's example.

Figure 24: View/edit multiple reports list sequence diagram.
The first six attributes are associated with UI view components existing in figure 21 (creating multiple reports view). These attributes must keep the reports chosen by the user. When the user presses the "create" button the methods used will be createMRView() and this one will use the getReportNorth(), getReportSouth(), getReportEast(), getReportWest(), getReportCenter() methods, to get the report's features in each position and put them, with the view's object help, in the database.

In the page which allows to view and editing of the multiple reports view (figure 24) the methods used are:

- The getAllMultipleReportsViews (idUser : long) return all views associated with the specified user id.
- The EditReportView (view: String) gets the view's features for editing.
- The removeReportView (view: String) removes, with view object help, the specified view.

### Figure 25: Multiple report view

```java
MultipleReportViewBean

- id
- name : String
- templateView : Set
- get(id)
- set(id)

view

MultipleReportViewBean

- name : String
- reportNorth : String = ""
- reportSouth : String = ""
- reportEast : String = ""
- reportWest : String = ""
- reportCenter : String = ""
- report
  - view : MultipleReportViewBean
    + createMRView() : void
    + getReportNorth() : String
    + getReportSouth() : String
    + getReportEast() : String
    + getReportWest() : String
    + getReportAvailable() : Arraylist
    + getReportCenter() : String
    + setReportNorth(rel : String) : void
    + setReportSouth(rel : String) : void
    + setReportEast(rel : String) : void
    + setReportWest(rel : String) : void
    + removeReportView(view : String) : void
    + editReportView(view : String) : void
    + getAllMRViews(id : long) : Arraylist
```
3.2.3 Report visualization

When the mouse pointer is over the report's visualization menu button, three options are displayed:

- Saved tasks (the use case is not considered in this FSPEC);
- Scheduled reports. Inside this option you have reports organized by technology;
- Multiple report view.

When the user clicks in a report (Ex: Report 1), some chart's or/and table's will be displayed depending on the template associated with it. In this case the template uses a table (west) and a report (center).

With the report presentation it appears another menu, in the top, having the following items:

- Group by (this use case is not considered in this FSPEC);
- Filtering (if the visualization has a table specified);
- Print;
- PDF creation.

![Web Reporting](image)

Figure 26: Report visualization view’s example
When user clicking in a report and in some cases, the report doesn’t have yet a template associated and then is presented the following view to him.

By clicking in “Create template” button user has access to the view where he can create a new template to associate with the report.

By clicking in “Select template” user has access to the view where he can choose a template to associate with the report.

By clicking in “Default template” a default template is associated with the report.

The selected report doesn’t have a template associated.

Choose one of this options.

Create template  
Select template  
Use default template

Figure 27: Report association template options view’s example.

Selecting the filtering option a popup will be displayed like it is showed in the figure below. In this situation the user must define the counter which will be applied the filter, the logical operator to make the compassion, the value and the color. Below user has access to the added filter list in which he can remove some filters. If he presses the “apply” button, all the filters will be applied to the table. If the user selects the check box only the rows selected by the filtering will be displayed.

Web Reporting

Report visualization (schedule report)

Welcome User

![Filter window](image)

Figure 28: Report visualization specifying a filter view’s example.
The figure below shows the filtering action result. The printing option will print the report and the PDF creation will export the report to a PDF file.

**Web Reporting**

*Report visualization: schedule report*

![Diagram of Web Reporting]

Figure 29: Report visualization filter result view's example.
When the user selects a multiple report's view, will be displayed a set of reports. In this case there will be displayed three reports.

Figure 30: Multiple reports view visualization view's example
If the user wants to add some filtering he must click in the filtering link. The process will be the similar as described in the filtering only a report but he must specify which report he wants to apply the filter.

Web Reporting

Report visualization.

![Web Reporting Diagram]

Figure 31: Multiple reports view visualization specifying a filter view's example.
Figure 32: Report visualization class diagram

In the simple report's visualization case there will be used the following methods:

The chooseReport(event: ActionEvent) handles action of choosing a report, getting the characteristics of the selected report like the name and using it to get the report id (getReportId) and then to get the associated template (getTemplate). If the report isn't in the database the WEB application must insert it and ask user to associate a template to it (figure 28).

Next step, the WEB application must confirm if exists an object serialization (getObjectSerialization) in the database (using reportBean from ReportBean class). Case exist, a precaution must be done, with the help of report attribute and using the sameReport method the WEB application must certify if the xml is the same from the object's serialization, through the date and time and then the view will be rendered. In case not, the xml data must be passed to objects. The report object (from Report visualization bean class) will be used and so his method newReport.

The newReport uses the addObject and the addCounter methods and when it finishes, the objectInstance created must be added to the ArrayList objectInstanceList.

The addObject creates a new ObjectInstanceBean object, with the help of the parserXML object, to get data for initializing the object.

The addCounter is used to add a counter to the objectInstance, using the addNewCounter and addCounterToList ObjectInstanceBean methods. The parameters to make possible this creation will be getting using the parserXML methods.

In the multiple reports visualization the ReportVisualizationBackingBean class must get the reports including in the multiple view, through the multipleRview object. Then the process described above must be done.

When the user wants to apply a filter, the applyFilter method will be used (including in the ReportVisualizationBackingBean class). When the user adds a new filter an object's FilterBean will be created and added to the filterList. If user clicks in the "apply" button, these objects will be applied to the tables. If the user clicks in the “cancel" button, these objects will be destroyed.
In the multiple reports visualization the report attribute of the FilterBean class must be introduced, to identify which report the filter will be applied to.

Figure 33: Report visualization sequence diagram
Figure 34: Report visualization specifying filters sequence diagram