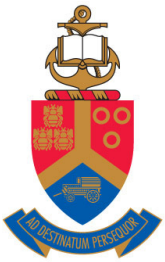

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS301 Mini Project

Round 4

Testing Report: Reporting module

Vivian Venter *u13238435*
Tienie Pritchard *u12056741*
Lindelo Mapumulo *u12002862*
Sphelele Malo *u12247040*
Martha Mohlala *u10353403*
Sean Hill *u12221458*
Goodness Adegbenro *u13046412*
Tsepo Ntsaba *u10668544*
Michael Nunes *u12104592*

Here's a link to GitHub.
`https://github.com/VivianLVenter/TestPhase-Reporting`

April 22, 2015

Contents

1 Testing TeamA

1.1 Functional Requirements Review

1.1.1 Treads.getThreadStats

The getThreadStats function has achieved what it needs to do but it has minimal error checking.The parameters that are required are all available.The set parameter contains a list of posts.The action parameter is tested for Num, MemCount, MaxDepth, AvgDepth.The action that is specified will carry out the appropriate action to be done with the threads such as calculate the number of threads.There is no error checking so if the action specified is not one of the ones in the functional requirements then the return value will be null which could cause a problem when the module is integrated with the other modules.There is no exception thrown so the function will exit without any knowledge of an error.

Full use case test rating

Partial Pass

The function will run but in some cases the lack of error checking will cause errors in integration.

1.1.2 GetThreadAppraisal

The GetThreadAppraisal function works and produces the required output and will calculate the appraisals.All the parameters and post conditions are present.The data is stored correctly into a dataset,JSON string, which contains all the entries for that thread and a action value.The action value which is specified as one of the parameters to be either Sum,Avg,Max,Min or Num.The entries contain all of the threads and the information about the thread including an ordinal vale for the thread which is used in the calculation that is specified by the action.There is no error checking so if the action value is not one that is specified then the action value will not be set and this could cause problems in integration.

Full use case test rating

Partial Pass

The function will run but in some cases the lack of error checking will cause errors in integration.

1.1.3 ExportThreadAppraisal

The ExportThreadAppraisal function could not be found.This function has a medium priority and is important to the system.No testing could be done on this use case.This functionality is important as it allows the user to export the thread appraisal and allows offline editing.

Full use case test rating

Failed

The function is missing so no testing.

1.1.4 Import Thread Appraisal

The ImportThreadAppraisal function could not be tested because it was not implemented.

Full use case test rating

Failed

The function is missing so no testing.

1.1.5 Export Thread

The ExportThread function was tested and runs successfully. It provides the functionality to backup the content of a thread or subset of a thread. It achieves this by getting the threads using the ThreadID parameter. It then converts the threads to a serialized object and writes the serialized thread object to a file for backup. However, this function does not throw exceptions when errors occur. It only sends an output to the console which will not be visible to the user. Therefore, there is no way of knowing if an error has occured.

Full use case test rating

Partial Pass

The function will run but in some cases the lack of error checking will cause errors in integration.

1.1.6 Import Thread

The ImportThread function was tested and runs successfully. It provides the functionality to restore the content of a thread or subset of a thread that was stored using the ExportThread function. It achieves this by retrieving the serialized thread that was exported to a backup file in the ExportThread function. It converts the serialized thread to an unserialized thread object and returns the imported thread. Therefore, this function meets all the pre and post condition requirements. This function however, does not throw appropriate exceptions if an error occurs to notify users of this error.

Full use case test rating

Partial Pass

The function will run but in some cases the lack of error checking will cause errors in integration.

1.2 Architectual Requirements Review

2 Testing TeamB

2.1 Functional Requirements Review

2.1.1 Threads.getThreadStats

2.1.2 Import Thread Appraisal

Data in an external file that was created using the exportThreadAppraisal function is used.

Partial Passed

The function do make use of a file. The file is sent as arguments in the importThreadAppraisal function (parameters are directory and fileName). So in this function they assume this is the file created by the exportThreadAppraisal function.

The data set is associated with only one member and only one specified appraisal.

Failed

The function never actually check if the data set is about only one member and only one specific appraisal. Therefore the data cannot be deemed as eligible for import.

A record contains all detail about the post along with a field that should contain an ordinal number that represents the levels of the specified appraisal.

Passed

The record does contain all detail about the post and a field that represents the leves of the specified appraisal which is the appraisalValue.

Edits to the data is ignored when importing a thread appraisal.

Failed

The function does not prevent edits to the data and there is not clear indicated of ignoring such edits.

For each record the assignAppraisalToPost function is applied.

Passed

The function does call assignAppraisalToPost for each record.

The appraisal level as stored in the file for each post is updated as an appraisal assigned by the member associated with the data set.

Failed

The function does not use the appraisal level to update the data set.

Check validity of member and appraisal.

Partial Passed

The function does check the validity of the member, however they have a dummy function that only returns true.
The function does check the validity of the appraisal, however no exception is thrown/raised it only returns true or false and there is also no means of catching an exception, that is the isValid function is not surrounded with try/catch blocks to catch exceptions and therefore the service delivery is not stopped when the appraisal is not valid.
The appraisal level is check for out of range, however no exception is thrown/raised when is it out of range and there the service delivery is not stopped if the appraisal level is out of range.

2.1.3 Export Thread

An external file is created.

Partial Passed

The function does create an external file, however :

- The data generated to by the queryThread function is assumed to be given as an argument to the function (parameter threadObject).

- The data is actually not correctly stored in the file since the threadObject is not parsed as a JSON-string. The object is directly stored in the file. So the actual data of the threadObject is not stored in the file.

2.1.4 Import Thread

An external file that was created by the exportThread function is used to restore the content of a thread.

Passed

The function does use an external file. Again the file is passed as arguments to the function (parameters directory and fileName), so they are assuming this file was generated by the exportThread function before importThread is called with the file as arguments.

The content of the external file is used to restore content of a thread/post.

Failed

The function never tries to restore or even store the content of the external file on the Buzz Space.

A post is added if and only if it is not in the Buzz Space already.

Failed

The function never adds the content to the Buzz Space and therefore does not check if it is already in the Buzz Space.

2.2 Architectual Requirements Review

Software quality requirements

- Reliability: The switch statement's (getThreadAppraisal.js) default case was not specified. If an error occurred, it wouldn't be traceable since there is no error handling. If "garbage" values were passed through this switch statement, the error's result could be a system failure.
- Maintainability: There is code duplication within the conditional statements. Many statistical operations exist and should someone wish to add them, it would be a daunting task. A function, instead of duplicated code, would simplify the modifications.
  These two files (importThread.js and importThreadAppraisal.js) also have duplicate code, the effects are similar to the ones mentioned earlier.
- Test-ability: Unit tests were included for each function. This reduces single-point-of-failure, because errors in smaller functions can be addressed without affecting the whole system.
- Audit-ability: (getThreadStats.js) has a few entries for auditing (ParentID, Author, Timestamp, Content, Status). These are not adequate for auditing, especially since there's a single timestamp (which was ambiguous).

Architectural components

- Framework(s): Node.js was used, as specified. Node-aop and Broadway plugin framework were included in "package.json" but they both appear unused.
- Technologies: JSON (JavaScript Object Notation) was used to store "thread objects". HTML5 was used, however, this was for testing the low-level implementation. Electrolyte was included in "package.json" but there were no signs of it's use for the actual implementation.

3 Conclusion

3.1 Conclusion