Testing Report

Group Name: Testing Status

Daniel Christopher Alves Araüjo	13073878
Taariq Ghoord	10132806
Lerato Molokomme	11197961
Semaka Malapane	13081129
Mpedi Mello	11210754
Ryno Pierce	12003922
Lutfiyya Razak	10198408
Frederick Snyman	13028741
Keagan Thompson	13023782

Git repository link:
`https://github.com/u13073878/`
`COS301-Testing-Status`

Date: 24 April 2015

Contents

1 assessProfile Use case

1.1 group A - assessProfile

1.2 group B - assessProfile

2 setStatusCalculator Use case

2.1 group A - setStatusCalculator

2.2 group B - setStatusCalculator

3 getStatusForProfile Use case

3.1 group A - getStatusForProfile

3.2 group B - getStatusForProfile

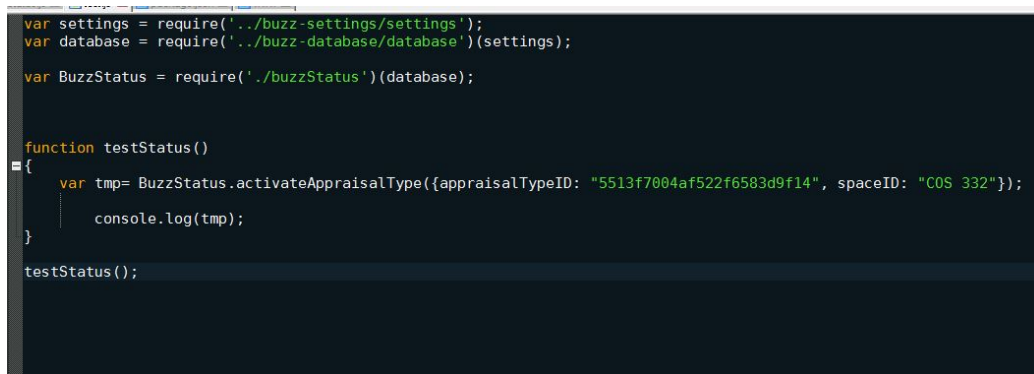# 4 createAppraisalType Use case

4.1 group A - createAppraisalType

4.2 group B - createAppraisalType

5 activateAppraisalType Use case

5.1 group A - activateAppraisalType

Status A provides an (activateAppraisalType) function, but does not match the specification. The function implemented asks for the (appraisalType ID), but does not ask for a specific time period to be activated as indicated in figure 45 of the master specification. The code for the unit test is given by the figure below:

```javascript
var settings = require('../buzz-settings/settings');
var database = require('../buzz-database/database')(settings);

var BuzzStatus = require('./buzzStatus')(database);


function testStatus()
{
    var tmp= BuzzStatus.activateAppraisalType({appraisalTypeID: "5513f7004af522f6583d9f14", spaceID: "COS 332"});

        console.log(tmp);
}

testStatus();
```
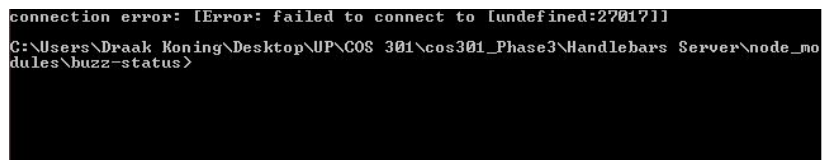
Figure 1: Unit test for activateAppraisalType

The following is the result of running the code:

```
connection error: [Error: failed to connect to [undefined:27017]]

C:\Users\Draak Koning\Desktop\UP\COS 301\cos301_Phase3\Handlebars Server\node_mo
dules\buzz-status>
```

Figure 2: Output of unit tests for activateAppraisalType

The first test, *activePeriodTest*, passes. *activePeriod()* takes two strings as parameters that represent the starting and ending date. However, it does no testing to ensure that the dates are in the correct format supported by JavaScript. Secondly, it does not have a callback parameter to specify a callback function, so it cannot be run asynchronously.

As you can see it could not connect to the database correctly in order to activate the appraisal type. Further inspection of the code show that it returns the value instead of using a callback function, this will cause it to always return empty or incorrect values as node runs asynchronously.

The implementation of the function does not meet the pre condition seeing that it doesn't test if the buzz space is active as well as, whether the active period is before the current date.

The post condition is net met due to the fact that the function could not connect to the database.

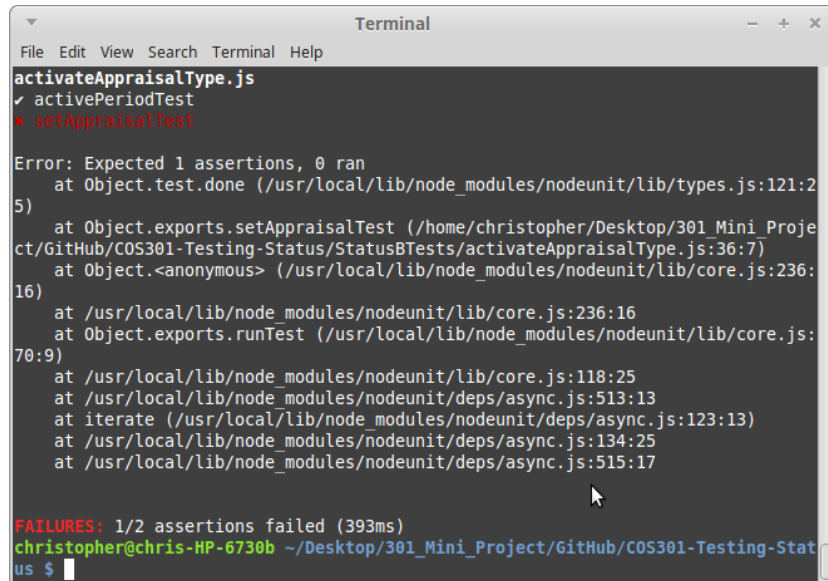## 5.2   group B - activateAppraisalType

Status B did not provide an explicit *activateAppraisalType* function that matches Figure 45 of the project specifications. However, Status B does provide two functions, *activePeriod()* and *setAppraisal()*.

The code for the unit tests is given by the figure below. For the test, NodeUnit was used.

```
9
10   /*
11       test for activePeriod
12    */
13   exports.activePeriodTest = function(test)
14   {
15       test.expect(1);
16
17       var expectedResult = JSON.stringify({"active_from":"2015-10-01","active_to":"2015-10-30"});
18       var result = status.activePeriod("2015-10-01", "2015-10-30");
19
20       test.equal(result, expectedResult);
21       test.done();
22   }
23
24   /*
25       test for setAppraisal
26    */
27   exports.setAppraisalTest = function(test)
28   {
29       test.expect(1);
30
31       var post_id = "0";
32       var appraisalName = "Test";
33
34       status.setAppraisal(post_id, appraisalName); //The function does not return any value
35
36       test.done();
37   }
```

Figure 3: Unit test for activateAppraisalType

The following was the output given by NodeUnit.

Figure 4: Output of unit tests for activateAppraisalType

The first test, *activePeriodTest*, passes. *activePeriod()* takes two strings as parameters that represent the starting and ending date. However, it does no testing to ensure that the dates are in the correct format supported by JavaScript. Secondly, it does not have a callback parameter to specify a callback function, so it cannot be run asynchronously.

The section function, *setAppraisal()*, fails. The function receives two parameters, a **post_id** and and **appraisal_name** to find a record in the database matching the **post_id** and assign the **appraisal_name** to **appraisal_id** in the database. The function has an error that attempts to assign a null value to **post.appraisal_id** in the database.

The pre-conditions are not tested in either function, although a assumption can be made about a buzz space being active. However, *setAppraisal()* does not test whether or not the active period is before the current date or time.

Due to the *setAppraisal()* failing, the function does not meet the post-condition of appraisalTypeAssignment being persisted to the database.

6 assignAppraisalToPost Use case

6.1 group A - assignAppraisalToPost

6.2 group B - assignAppraisalToPost