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

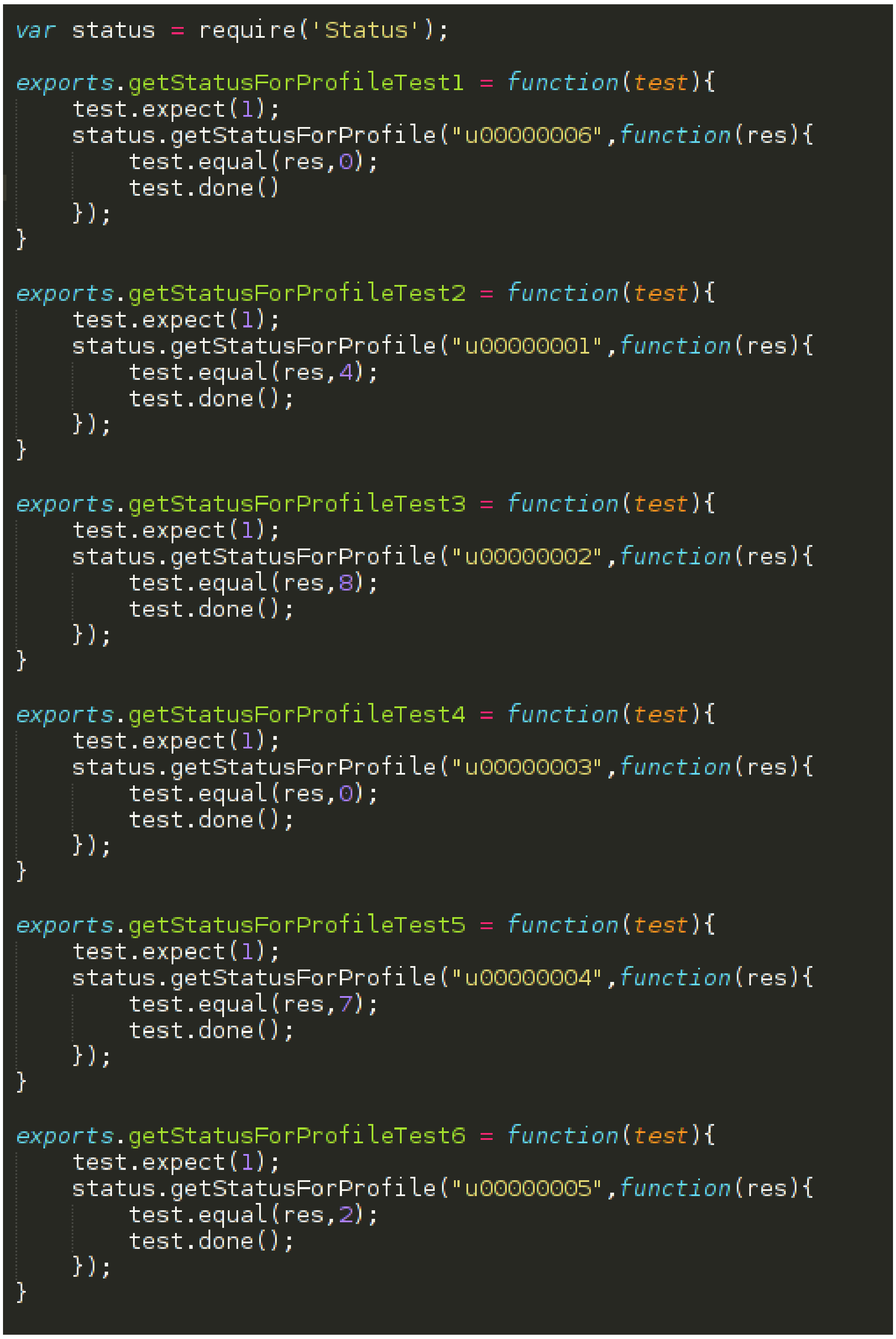
3.1 group A - getStatusForProfile

3.2 group B - getStatusForProfile

Status B provided a *getStatusForProfile* as in Figure 41 of the master specifications. No pre- or post-conditions are tested for in the code, however none were provided in the given master specifications.

The function that Status B provided takes as parameters the ID of the user being queried.

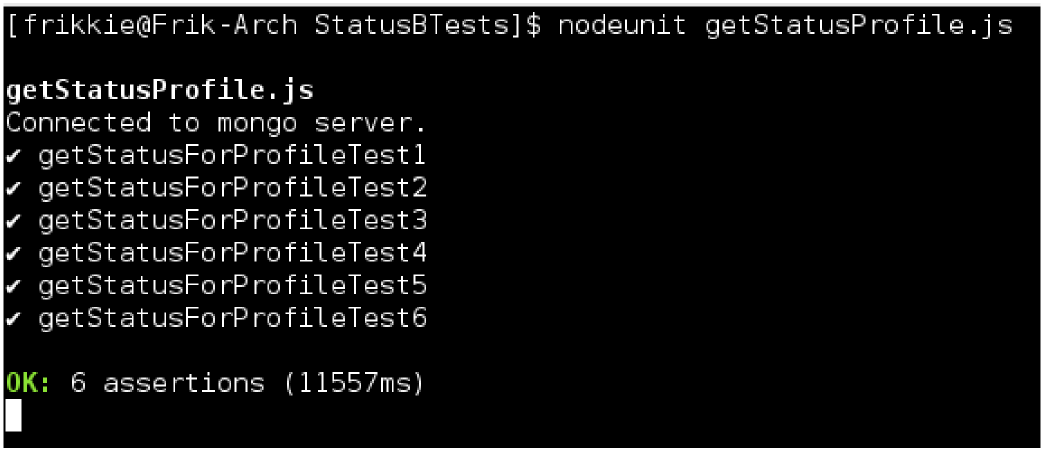
Nodeunit was used for doing unit tests on this code, and the code of the unit tests appears in the figure below:



```
var status = require('Status');

exports.getStatusForProfileTest1 = function(test){
    test.expect(1);
    status.getStatusForProfile("u00000006",function(res){
        test.equal(res,0);
        test.done()
    });
}

exports.getStatusForProfileTest2 = function(test){
    test.expect(1);
    status.getStatusForProfile("u00000001",function(res){
        test.equal(res,4);
        test.done();
    });
}

exports.getStatusForProfileTest3 = function(test){
    test.expect(1);
    status.getStatusForProfile("u00000002",function(res){
        test.equal(res,8);
        test.done();
    });
}

exports.getStatusForProfileTest4 = function(test){
    test.expect(1);
    status.getStatusForProfile("u00000003",function(res){
        test.equal(res,0);
        test.done();
    });
}

exports.getStatusForProfileTest5 = function(test){
    test.expect(1);
    status.getStatusForProfile("u00000004",function(res){
        test.equal(res,7);
        test.done();
    });
}

exports.getStatusForProfileTest6 = function(test){
    test.expect(1);
    status.getStatusForProfile("u00000005",function(res){
        test.equal(res,2);
        test.done();
    });
}
```

Figure 1: Unit tests for getStatusForProfile

Executing the tests above resulted in the following output:



```
[frikkie@Frik-Arch StatusBTests]$ nodeunit getStatusProfile.js

getStatusProfile.js
Connected to mongo server.
✔ getStatusForProfileTest1
✔ getStatusForProfileTest2
✔ getStatusForProfileTest3
✔ getStatusForProfileTest4
✔ getStatusForProfileTest5
✔ getStatusForProfileTest6

OK: 6 assertions (11557ms)
```

Figure 2: Results of the unit tests for getStatusForProfile

The tests that are run, checks the status value for six users present in the database, with differing status values. All six test pass successfully and the function *getStatusForProfile* appears to hav been implemented correctly as per the master specification. These test cases cover 100% of the use cases as defined in the master specification.

4 createAppraisalType Use case

4.1 group A - createAppraisalType

4.2 group B - createAppraisalType

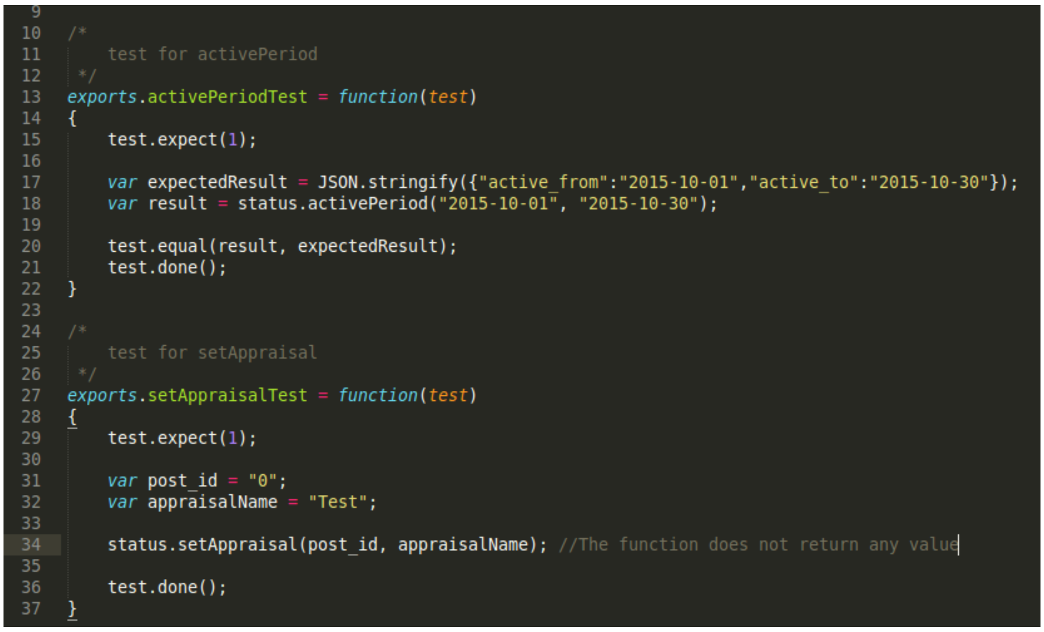
5 activateAppraisalType Use case

5.1 group A - activateAppraisalType

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()*.

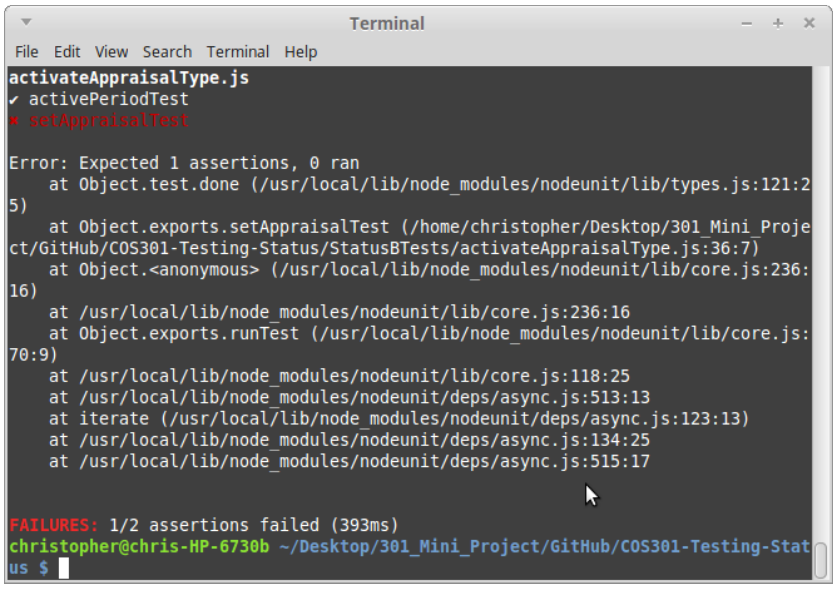
The code for the unit tests is given by the figure below. For the test, NodeUnit was used.



```
/*
    test for activePeriod
 */
exports.activePeriodTest = function(test)
{
    test.expect(1);

    var expectedResult = JSON.stringify({"active_from":"2015-10-01","active_to":"2015-10-30"});
    var result = status.activePeriod("2015-10-01", "2015-10-30");

    test.equal(result, expectedResult);
    test.done();
}

/*
    test for setAppraisal
 */
exports.setAppraisalTest = function(test)
{
    test.expect(1);

    var post_id = "0";
    var appraisalName = "Test";

    status.setAppraisal(post_id, appraisalName); //The function does not return any value

    test.done();
}
```

Figure 3: Unit test for activateAppraisalType

The following was the output given by NodeUnit.



```
activateAppraisalType.js
✔ activePeriodTest
✖ setAppraisalTest

Error: Expected 1 assertions, 0 ran
    at Object.test.done (/usr/local/lib/node_modules/nodeunit/lib/types.js:121:2
5)
    at Object.exports.setAppraisalTest (/home/christopher/Desktop/301_Mini_Proje
ct/GitHub/COS301-Testing-Status/StatusBTests/activateAppraisalType.js:36:7)
    at Object.<anonymous> (/usr/local/lib/node_modules/nodeunit/lib/core.js:236:
16)
    at /usr/local/lib/node_modules/nodeunit/lib/core.js:236:16
    at Object.exports.runTest (/usr/local/lib/node_modules/nodeunit/lib/core.js:
70:9)
    at /usr/local/lib/node_modules/nodeunit/lib/core.js:118:25
    at /usr/local/lib/node_modules/nodeunit/deps/async.js:513:13
    at iterate (/usr/local/lib/node_modules/nodeunit/deps/async.js:123:13)
    at /usr/local/lib/node_modules/nodeunit/deps/async.js:134:25
    at /usr/local/lib/node_modules/nodeunit/deps/async.js:515:17


FAILURES: 1/2 assertions failed (393ms)
christopher@chris-HP-6730b ~/Desktop/301_Mini_Project/GitHub/COS301-Testing-Stat
us $
```

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