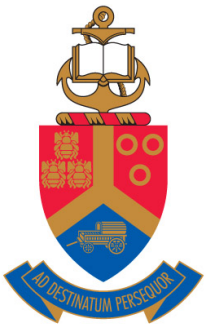

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

2017 COS 301 Project
Testing Documentation

Seonin David
Joshua Moodley
Jacques Smulders
Jordan Daubinet
Nicaedin Suklul

Contents

1 Presentation Tier **3**
1.1 Introduction . . . 3
1.2 About the test . . . 3
1.3 Pre-test information . . . 3
1.4 The test . . . 3
1.5 Login Page scenario's . . . 3
1.6 Project Creation Page scenario's . . . 3
1.7 Employee Register Page Scenario's . . . 4
1.8 Project dashboard page scenario's . . . 4

2 Server Tier **4**
2.1 Introduction . . . 4

3 Data Tier **5**
3.1 Introduction . . . 5
3.2 Scalability testing . . . 5

4 AI Tier **5**
4.1 Introduction . . . 5

5 MochaJS Tests **6**
5.1 Test 1 . . . 6
5.1.1 Explanation . . . 6
5.1.2 Test Code . . . 6
5.1.3 Test results . . . 6
5.1.4 Comment . . . 6
5.2 Test 2 . . . 6
5.2.1 Explanation . . . 6
5.2.2 Test Code . . . 6
5.2.3 Test results . . . 6
5.2.4 Comment . . . 6
5.3 Test 3 . . . 7
5.3.1 Explanation . . . 7
5.3.2 Test Code . . . 7
5.3.3 Test results . . . 7
5.3.4 Comment . . . 7
5.4 Test 4 . . . 7
5.4.1 Explanation . . . 7
5.4.2 Test Code . . . 7
5.4.3 Test results . . . 7
5.4.4 Comment . . . 8

6 Benchmark Tests **8**
6.1 Artillery testing framework . . . 8
6.2 Loadtest testing framework . . . 8

1 Presentation Tier

1.1 Introduction

The presentation tier testing covers all user interaction with the systems front end presentation.

1.2 About the test

The format of this test is a usability test which targets the systems design. A usability test is carried out as follows: users are given a set of scenarios and tasks to complete. In each scenario a user attempt to complete a task is timed and feedback both from the test itself and the user carrying out the scenario are taken. The scenario's are created to test the functionality of the application and to record the user experience well using the product.

1.3 Pre-test information

The users targeted to participate in the test are made up of two phases. The first phase is a group of computer science students who are familiar with the concept of interaction design and the purpose of the test was to get feedback on design issues and improvements. The second phase is a group of kpmg staff members who in the future would use the project management application for its purpose of creation and the purpose of the test was to get feedback on its functionality and to investigate adaptions to the current functionality or added functionality that has to be made. All users sign a consent form and are thanked thoroughly afterwards for their participation.

1.4 The test

For the first phase, users where given access to login credentials for all three user types (administrator, project manager and employee) with a system that was pre populated with dummy data. They were given free reign to do what they want with it in an environment that did not cont role their actions. Each user had a project member besides them to answer any questions needed and were encouraged to give as much feedback as possible about any aspect of the project and its presentation.

The second phase followed a more orthodox usability test approach. Users where not given any information about the product and project team members could not answer any questions regarding the product. Users where given a set of scenarios to follow and complete. The scenarios were created to target the different functionality of the application. Each user was timed by a project team member with each scenario, based on how long it took them to complete it. At the end of each scenario users where encouraged to give feedback before moving onto to the next scenario. The results of the test are used to make improvements for reducing the time it takes to performa each scenario as well as increasing the user experience with each scenario.

With each scenario the user was given a laptop which was open to the intended web page.

1.5 Login Page scenario's

- From the login page, use the user name "emp1" and password "test" to login to the application.
- From the login page locate the option for lost my password and using the user name "emp2" request for a new password.

1.6 Project Creation Page scenario's

- From the Project Creation page, request to allocate 4 employees with the skill tags: data analysis, security and MS office. It must have the title "KPMG" and must run from August the first this year to September the first this year. Its Budgest must be R400 000. The project Owners details are as follows: Project Owner "Owner1", email "owner1@gmail.com", contact number "000-000-000". The project managers information is as follows: Project Manager "Manager1", email "manager1@gmail.com", contact number "111-111-111".

- From the Project Creation page, request to allocate 2 employees with the skill tags: data auditor. request to change one of the given employees to another from the recommended list.
- From the Project Creation page, request to allocate 2 employees with the skill tags: data auditor. The create a new project with the given details. It must have the title "KPMG" and must run from August the first this year to September the first this year. Its Budgest must be R400 000. The project Owners details are as follows: Project Owner "Owner1", email "owner1@gmail.com", contact number "000-000-000". The project managers information is as follows: Project Manager "Manager1", email "manager1@gmail.com", contact number "111-111-111".

1.7 Employee Register Page Scenario's

- From the Register page, register a new administrator with the details of your choice.
- From the Register page, register a new Manager with the details of your choice.
- From the Register page, register a new Employee with the details of your choice.

1.8 Project dashboard page scenario's

- From the Project Dashboard page, view the KPMG project.
- From the Project Dashboard page, extend the date of the KPMG project.
- From the Project Dashboard page, write down the team members names who are currently in the UP project as well as the status of the project and its progress rating.
- From the Project Dashboard page, find a way to create a new project.

2 Server Tier

2.1 Introduction

The Logic tier testing covers all the current Server functionality. The tests are as follows:

- Checking that the current user is an authentic user which is currently stored in the database.
- Searching for a user in the database based on their id/username and returning the json object which describes all the user's attributes, using the findUsers function.
- Searching for a user in the database based on their id/username and returning the json object which describes the users role, using the findUsers function.
- Searching for all users in the database which have a defined attribute and its corresponding value and returning the json object which holds all these users, using the findUsers function.
- Creating a new project with all values stored in the given json parameter and storing it into the database, using the insertProject function.
- Searching for a project with the given Id in the database and returning the json object which describes all the projects attributes, using the findProjects function.
- Creating a hashed key for a given value and returning the new hashed key value, using the encrypt function.
- Authenticating that a user with a given id and password is currently stored in the database, using the authenticate function.

3 Data Tier

3.1 Introduction

The data tier testing covers all the current database functionality. The tests are as follows:

- Creating an employee json object with all values stored in the given json parameter and adding it to the database using the insertUser function.
- Searching for a user in the database based on their id/username and returning the json object which describes all the user's attributes, using the findUsers function.
- Searching for a user in the database based on their id/username and returning the json object which describes the users role, using the findUsers function.
- Searching for all users in the database which have a defined attribute and its corresponding value and returning the json object which holds all these users, using the findUsers function.
- Creating a new project with all values stored in the given json parameter and storing it into the database, using the insertProject function.
- Searching for a project with the given Id in the database and returning the json object which describes all the projects attributes, using the findProjects function.
- Creating tasks and milestones for a project using the insertTask and insertMilestone functions respectively, and using the findProjects functions to ensure that the tasks and milestones are referenced appropriately.
- Assigning an employee to a project using the AssignProject function, and using the findProjects and findUsers functions to ensure that the project and employee are referencing one other correctly.
- Creating a hashed key for a given value and returning the new hashed key value, using the encrypt function.
- Authenticating that a user with a given id and password is currently stored in the database, using the authenticate function.

3.2 Scalability testing

To test the data layer on a larger scale the following functions where used:

- Creating 5 test employees and inserting them into the database, using the createTestEmployees function.
- removing all employees which are currently stored in the database, using the removeTestEmployees function.
- Displaying all users which are currently stored in the database to the console, using the viewEmployees function.

4 AI Tier

4.1 Introduction

5 MochaJS Tests

5.1 Test 1

5.1.1 Explanation

The first test that we wrote was to test that the login page renders.Thus the test should pass if it receives a code 200 indicating that the page rendered properly.

5.1.2 Test Code

```
describe('Login Page render test', function() {
it("renders successfully", function(done) {
request(app).get('/').expect(200, done);
})

});
```

5.1.3 Test results

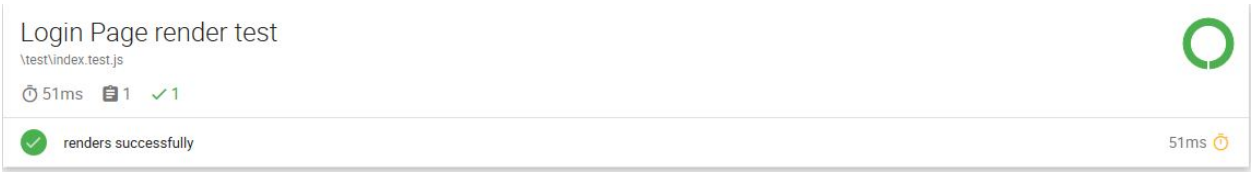

Figure 1: Login page render test

5.1.4 Comment

In Figure 1 it can be seen the test passed.Therefore the login page renders and there are no errors with the login page.

5.2 Test 2

5.2.1 Explanation

This tests checks whether data can be sent to the login page without any errors.

5.2.2 Test Code

```
request(app)
.get('/')
.set('Accept', 'application/json')
.set('Content-Type', 'application/json')
.send({ username: 'Seonin', password: 'Seonin' })
.expect(200)
.end(done);
```

5.2.3 Test results

5.2.4 Comment

Figure 2 shows that the test passes because it can send the object to the page and the page will still render.

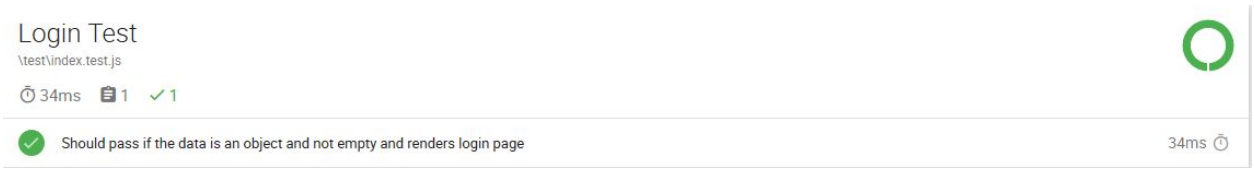

Figure 2: Login test

5.3 Test 3

5.3.1 Explanation

This test will try to route to a page that does not exist and expect a 404 page not found error code.

5.3.2 Test Code

```
request(app).get('/unknown_page').expect(404, done);
```

5.3.3 Test results

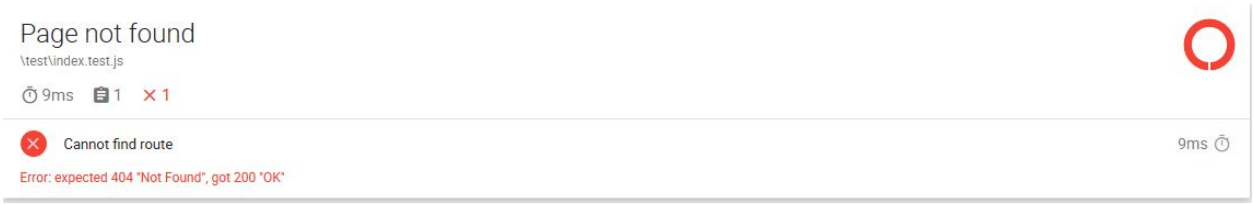

Figure 3: Error 404/200: Page not found error with pass on routing

5.3.4 Comment

In Figure 3. This test will fail because even though the page does not exist. We put a 404 page that will display if a page is not found.Thus it gives code 200 because it routes to the error page.

5.4 Test 4

5.4.1 Explanation

This test expects 200 status because it should route to the error page whenever there is an error 404 code.

5.4.2 Test Code

```
request(app).get('/unknown_page').expect(200, done);
```

5.4.3 Test results

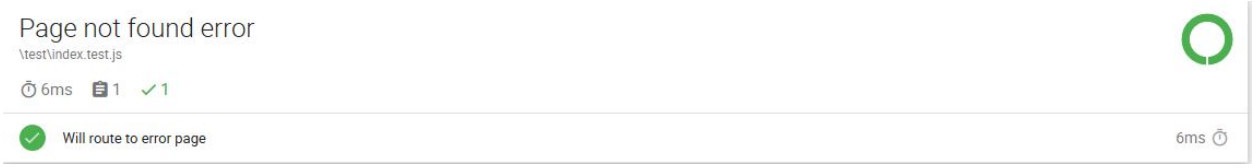

Figure 4: Error 404: Page not found error

5.4.4 Comment

In Figure 4 it passes the test because a page was inserted to route to if there is an error 404 code.

6 Benchmark Tests

6.1 Artillery testing framework

For benchmark test run on Seonin David's laptop the following results were captured:

- 600 scenarios were created in 70 s
- All 600 scenarios returned with code 200 which means that all the requests did not produce and errors.
- Min latency of 6 ms
- Max latency of 254.8 ms
- Mean latency of 118.7 ms

6.2 Loadtest testing framework

- Target URL: **https://localhost:4000/**
- Time: 50s
- Concurrency level: 100
- Requests per second: 140
- Completed requests: 10175