

Project - ViaSofie.be

Test plan

Bachelor Applied Informatics

Year 2 Semester 1-2

Academic year 2015 – 2016

Tom Bruyninx

Tim Cocx

Glenn Michiels

Olivier Van Bulck

Michaël Vanderborght

In cooperation with

Sofie Andriesen – ViaSofie.be

17 juni 2016

**Test plan**

ViaSofie.Be

Table of Contents

[1 Introduction 4](#_Toc453957525)

[1.1 Test plan summary 4](#_Toc453957526)

[1.2 Project description 4](#_Toc453957527)

[1.2.1 Functionalities 5](#_Toc453957528)

[1.3 Sponsor 7](#_Toc453957529)

[1.3.1 Approved by 7](#_Toc453957530)

[1.3.2 Sponsored by 7](#_Toc453957531)

[1.4 Test strategy 7](#_Toc453957532)

[1.4.1 Exploratory tests 8](#_Toc453957533)

[1.4.2 Usability - & UAC tests 8](#_Toc453957534)

[1.4.3 Security tests 8](#_Toc453957535)

[1.4.4 Multi-platform - & multi-browser tests 9](#_Toc453957536)

[1.4.5 Stress/ load tests 9](#_Toc453957537)

[1.4.6 Software/ hardware requirements 9](#_Toc453957538)

[1.4.7 Risks 10](#_Toc453957539)

[1.4.8 Responsibilities and restrictions 10](#_Toc453957540)

[1.5 List of tests 10](#_Toc453957541)

[1.6 Schedule 11](#_Toc453957542)

[2 Exploratory tests 12](#_Toc453957543)

[2.1 Activities 12](#_Toc453957544)

[2.2 Resources 13](#_Toc453957545)

[2.3 Tools 13](#_Toc453957546)

[2.4 Documentation 13](#_Toc453957547)

[2.5 Code coverage 13](#_Toc453957548)

[2.6 Results 14](#_Toc453957549)

[3 Usability - & UAC tests 14](#_Toc453957550)

[3.1 Activities 14](#_Toc453957551)

[3.2 Resources 14](#_Toc453957552)

[3.3 Tools 14](#_Toc453957553)

[3.4 Code coverage 15](#_Toc453957554)

[3.5 Results 15](#_Toc453957555)

[4 Security tests 15](#_Toc453957556)

[4.1 Activities 15](#_Toc453957557)

[4.2 Resources 15](#_Toc453957558)

[4.3 Tools 15](#_Toc453957559)

[4.4 Code coverage 15](#_Toc453957560)

[4.5 Results 15](#_Toc453957561)

[5 Multi-platform - & multi-browser tests 16](#_Toc453957562)

[5.1 Activities 16](#_Toc453957563)

[5.2 Resources 16](#_Toc453957564)

[5.3 Tools 16](#_Toc453957565)

[5.4 Code coverage 17](#_Toc453957566)

[5.5 Results 17](#_Toc453957567)

[6 Stress/load tests 17](#_Toc453957568)

[6.1 Activities 17](#_Toc453957569)

[6.2 Resources 17](#_Toc453957570)

[6.3 Tools 17](#_Toc453957571)

[6.4 Code coverage 17](#_Toc453957572)

[6.5 Results 17](#_Toc453957573)

[7 Conclusion 18](#_Toc453957574)

[8 Appendix 19](#_Toc453957575)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Documentmanagement | | | | |
| Version | **Date** | **Distribution** | **Status** | **Major changes** |
| 0.1 | 06/06/2016 | Intern | Conceptual | Test Strategy |
| 0.2 | 10/06/2016 | Intern | Conceptual | Test Analysis |
| 0.3 | 15/06/2016 | Intern | Conceptual | Spellcheck |
| 1.0 | 15/06/2016 | Intern  Kelly Casal | Finished | Finished test plan |

# Introduction

## Test plan summary

The test plan is an essential piece of our development process. This document serves as a guide to the whole team regarding functionality development. Being the backbone of all testing related activities in the production lifecycle, the whole team must have a good understanding of what this entails.

Our plan will consist of several parts describing the environment, the functionalities to be tested, the different testing techniques we will use and the general planning of testing throughout the project. Testing is essential to reveal potential issues in an early stage.

To prevent tunnel vision and design flaws in both layout and functionality we will test our application extensively during the development phase. Near the end of the project a complete week will be reserved for testing purposes. As a result, we hope our application meets all required conditions.

## Project description

This project consists of designing, building and developing an interactive web application for a start-up real estate agent. The application will provide a website as well as an online platform for the different stakeholders. This platform will feature an easy way to show the complete real estate document required for the sale of a building.

Above all we need to design a nice, attractive and appealing website. The design is one of the centre pieces of this project. The website will serve as the online promo sign for Ms. Andriesen’s services. We aim for a “less is more”, visual and professional design.

Come second, all functionalities included should work and be easy to use. The platform should provide an intuitive lay-out. As a result, this should help the adoption rate and usage of our application.

Since Ms. Andriesen does not have any technical knowledge, it must be possible to manage by using an interface. There should be no coding or any other website development knowledge required to change data on the website. We will include an admin panel for this. In this panel the site administrator will be able to add real estate documents, buildings for sale, status updates, …

Next to these core functionalities the website will also feature contact forms. The site will act as an easy communication channel. For both new clients and sellers.

Our project rests on 3 cornerstones: intuitive, appealing design and easy-to-use.

### Functionalities

These functionalities will be included in the application and need to be tested:

#### Public site

|  |  |  |
| --- | --- | --- |
| Functionalities on the website | | |
| Functionality | **Parts** | **Tests** |
| Homepage | 1. Navigation 2. Responsiveness 3. Keyword search 4. Most popular houses 5. E-book request | Exploratory Tests  Usability Tests  User Acceptance Tests  Multi-platform Tests  Multi-browser Tests |
| Homes overview | 1. Navigation 2. Responsiveness 3. Functionality 4. Keyword search 5. Advanced search 6. Back button | Exploratory Tests  Usability Tests  User Acceptance Tests  Multi-platform Tests  Multi-browser Tests |
| Homes detail view | 1. Navigation 2. Responsiveness 3. Lay-out 4. Information | Exploratory Tests  Usability Tests  User Acceptance Tests  Multi-platform Tests  Multi-browser Tests |
| FAQ | 1. Navigation 2. Responsiveness 3. Text layout 4. Tabs | Exploratory Tests  Usability Tests  User Acceptance Tests  Multi-platform Tests  Multi-browser Tests |
| Contact | 1. Navigation 2. Responsiveness 3. Contact form 4. Google maps | Exploratory Tests  Usability Tests  User Acceptance Tests  Multi-platform Tests  Multi-browser Tests |
| About us | 1. Navigation 2. Responsiveness 3. Text layout 4. Tabs | Exploratory Tests  Usability Tests  User Acceptance Tests  Multi-platform Tests  Multi-browser Tests |
| Disclaimer / privacy | 1. Navigation 2. Responsiveness 3. Text layout | Exploratory Tests  Usability Tests  User Acceptance Tests  Multi-platform Tests  Multi-browser Tests |
| Language popup | 1. Popup 2. Language button 3. Cookie 4. Cookie notification | Exploratory Tests  Usability Tests  User Acceptance Tests  Multi-platform Tests  Multi-browser Tests |

#### Seller platform

|  |  |  |
| --- | --- | --- |
| Functionalities on seller platform | | |
| Functionality | **Parts** | **Tests** |
| Login | 1. Login authentication 2. Error popup’s | Exploratory Tests  Security Tests  Multi-platform Tests  Multi-browser Tests |
| Overview | 1. Navigation 2. Responsiveness | Exploratory Tests  Usability Tests  User Acceptance Tests  Multi-platform Tests  Multi-browser Tests |
| Detail view | 1. Navigation 2. Responsiveness 3. Lay-out 4. Information completion | Exploratory Tests  Usability Tests  User Acceptance Tests  Multi-platform Tests  Multi-browser Tests |

#### Admin panel

|  |  |  |
| --- | --- | --- |
| Functionalities on the admin panel | | |
| Functionality | **Parts** | **Tests** |
| Login | Login authentication  Error popup’s | Exploratory Tests  Security Tests  Multi-platform Tests  Multi-browser Tests |
| Overview | Navigation  Responsiveness  Login authentication | Exploratory Tests  Usability Tests  User Acceptance Tests  Security Tests  Multi-platform Tests  Multi-browser Tests |
| Users | Add users  Manage users  Remove users  Grouped actions | Exploratory Tests  Usability Tests  User Acceptance Tests  Security Tests  Multi-platform Tests  Multi-browser Tests |
| Addresses | Add address  Manage addresses  Remove addresses  Link address to user/house | Exploratory Tests  Usability Tests  User Acceptance Tests  Security Tests  Multi-platform Tests  Multi-browser Tests |
| Houses | Add houses  Manage houses  Remove houses  Add pictures to house  Add contract to house | Exploratory Tests  Usability Tests  User Acceptance Tests  Security Tests  Multi-platform Tests  Multi-browser Tests |
| Houses /  Real estate contract | Add contract  Add/remove status to house  Add/remove contract documents  Manage contract | Exploratory Tests  Usability Tests  User Acceptance Tests  Security Tests  Multi-platform Tests  Multi-browser Tests |
| Slideshow | Add pictures  Remove pictures | Exploratory Tests  Usability Tests  User Acceptance Tests  Security Tests  Multi-platform Tests  Multi-browser Tests |

## Sponsor

### Approved by

1. Tom Bruyninx
2. Glenn Michiels
3. Tim Cocx
4. Olivier Van Bulck
5. Michaël Vanderborght

### Sponsored by

* Ms. Andriesen
* ViaSofie.be
* ViaSofie Development Team 3
* Kelly Casal Mosteiro
* Marijn De Pooter

## Test strategy

As pointed out in the introduction, testing is an essential part of our development process. Our whole application will have to pass several tests to ensure all required functionalities. The platform consists of pages which contain information. It’s very important system users can access this information easily. Next to these pages there will also be several functionalities like contact forms, logins, … available on the platform. It’s equally important that these functionalities work as intended.

For this project we have selected the following tests:

* Exploratory tests
* Usability tests & user acceptance tests
* Security tests
* Multi-platform tests & multi-browser tests
* Unit tests
* Stress/load test

### Exploratory tests

Exploratory tests will be used to test the overall functionality of the site. By running through the platform unscripted, most of the functions will be used as they would be when the platform is online. These tests are time intensive and have to be done manually but they can be done without any preparations.

The key to exploratory testing is writing down all the performed actions. This will show a logical sequence representing the activity/sequence diagram made during analysis. By building the platform along this sequences, it will ensure that every part is included properly and works as it should.

#### Pass criteria

Exploratory testing is in essence running through the application unscripted and checking all functionalities. These tests will be done throughout development by every member of the development team.

The “pass conditions” for exploratory testing are defined by the tester at the moment of testing. But they will cover “general working of the functionality” as well as “listing bugs”.

### Usability - & UAC tests

Usability tests will be used to check if our application solves the problem. By setting up several scenarios which will be proposed to system users. We will be able to check if our application delivers the necessary support for all required actions by the system user.

User Acceptance Tests (UAC) are a different side of the same coin. We will use the same scenario’s as for usability tests but rather than checking if our application works the way it should, we will be checking whether our application meets the pre-set requirements in the analysis.

#### Pass criteria

The Pass criteria will be defined in several separate scenarios to test each functionality. Usability tests will be run by the development team itself. The scenarios will be written by one team member, while other team members execute these scenarios.

For UAC tests we will use the same scenarios as for usability tests. However, these scenarios will be presented to real system users. (Ms. Andriesen). Since no real test audience is present due to the organisation being a start-up. 1TI students will test the application as if they were real system users.

### Security tests

Security tests are necessary for checking the implementation of our login. The administration panel and the seller contract platform are protected by login. It’s essential to check if these pages are really protected by our security measures. Apart from the login we will also configure the necessary security measures on the provided VPS server. These measures only apply to the provided VPS server by AP hogeschool.

#### Pass criteria

Security tests will be coupled with unit tests since we will focus on login and authentication. To pass, all pages protected by the login should be made unavailable if not logged in.

The server security tests will comprise of certificate checks (the https certificate should be in order) and port checks. Only the necessary ports should allow connections.

### Multi-platform - & multi-browser tests

We’re building a web application. This means our application should be optimized for different platforms. Rescaling of the UI interface should be tested on different platforms.

Apart from multiple platforms our application should also be tested in different browsers. Different browsers render the UI in different ways. We aim for a universal UI in all major browsers

#### Pass criteria

The UI of the application should be universal over different browsers. It should also scale properly over different platforms. These tests will be run together as each browser will also be tested on each platform.

* Rescaling over different platforms
* UI in different browsers

### Stress/ load tests

Stress/load tests will be performed to check if the provided infrastructure can keep up with the required system resources of the website. The webserver should be able to manage more than a couple users. This test will be run on the development environment provided by the AP hogeschool. It will not resemble the real life scenario in which the website will be hosted by a hosting service.

#### Pass criteria

The criteria for stress and load tests are determined by the whole development team. The response time of the server has to be reasonably fast depending on location/ internet connection and client.

The server shouldn’t go down by several users trying to do actions at the same time. So as long as the server stays online the test is passed.

### Software/ hardware requirements

To run all scheduled tests we require certain hardware and software to be used. This list is not limited to these requirements and can be expanded/altered at all times.

**Hardware requirements:**

* Public available development environment/current version
* Client to access the server
* Access to the actual deployment environment

**Software requirements:**

* Browsers: Google Chrome, Mozilla Firefox, Internet Explorer, Microsoft Edge
* iPad/Mac with Safari browser
* Django Python framework
* Selenium browser automation
* Loadimpact.com stresstest

### Risks

Testing our application implicates the possibility of not being able to run certain tests. The most commonly possible risks posed to our testing activities have been listed:

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Chance  1-5 | Impact  1-5 | Action |
| No internet connection | 1 | 5 | Use public hotspot to sync latest GitHub commit.  Run the application locally |
| Client crash | 2 | 4 | Search other client |
| Development server not available. | 4 | 2 | Run the application locally |
| Forgot to note results | 1 | 5 | Rerun tests |
| Software not available | 2 | 4 | Search for other software |
| Bug in used packages | 1 | 5 | Wait for solution |
| The test subjects are not available | 4 | 3 | Make new appointment/new tests subject |
| Peer tutoring students test results not reliable | 3 | 2 | Development team runs the tests again |
| Development team can’t run tests | 1 | 4 | Reassign tests to different team member. (Conference call with unavailable team member) |
| Development incompatibility with tests | 2 | 5 | Rewrite the test scenario |
| New change that makes test obsolete | 3 | 2 | Write new tests |
| Delayed test results | 3 | 4 | Give notice to the team to hurry up. |
| IDE not available due to large updates | 1 | 3 | Delay the timing of tests. |
| Platform/browser not available | 2 | 4 | Wait till the platform is available. Search for other possibilities |
| Bug in browser | 1 | 5 | Wait for solution |

### Responsibilities and restrictions

Regarding testing the responsibilities of each team member depend on which test is run. For most tests the scenarios were written by Tom and discussed by the whole team. If the scenario and criteria are approved the test can be completed.

The initial scenarios for usability tests and UAC tests were written by our peer tutoring groups. Their cases were corrected by Tom, Tim and Michael. Before being submitted for approval.

In case of exploratory testing, each team member is responsible for setting the pass criteria. The test results are discussed by the whole team for approval.

## List of tests

Check 1.2.1.1 – 1.2.1.2 – 1.2.1.3 for an overview of each functionality and each test to be

## Schedule

|  |  |  |
| --- | --- | --- |
| Test | Timeline | Done by |
| Exploratory tests | Every sprint | All team members |
| Usability tests | 09/05 – 29/05 | Peer tutoring students |
| UAC tests | 23/06 | Sofie Andriesen |
| Security tests | 13/06 – 17/06 | Tom |
| Multi-platform tests  Multi-browser tests | Every sprint | All team members |
| Stress/load tests | 13/06 – 17/06 | Tom / Tim |

# Exploratory tests

## Activities

|  |  |
| --- | --- |
| Functionality | Activities |
| Navigation\* | Top-Menu/ Bottom-Menu:   1. Click link 2. Check if the redirection is correct 3. Note the results (if negative) |
| Page content\* | Checking lay-out and content   1. Open page 2. Note down every suggestion 3. Note down possible bugs 4. Discuss with development team |
| Keyword search | Test input/ output   1. Fill in test data 2. Click search 3. Check if search data is shown correctly |
| Advanced search | Test criteria selection/ output   1. Fill in test criteria to match 1 certain house 2. Click search 3. Check if search data is shown according to criteria |
| Language popup | Setting the language   1. Open the website 2. Set the language (is the popup shown?) 3. Refresh the page (F5 and ctrl+F5) 4. Check if the chosen language is retained. (No popup shown) |
| Contact forms\*\* | Sending messages   1. Open the contact / sales platform page 2. Fill in the contact form  (Correct information and incorrect information) 3. Check the CAPTCHA 4. Send the message 5. Control any possible error messages |
| Login | Login for admins   1. Click login 2. Fill out the form 3. Check redirection to admin panel   Login for house sellers   1. Click login 2. Fill out the form 3. Check redirection to house sell panels |
| Admin – Add users | Log in to the admin panel   1. Click “users” 2. Click “add user” 3. Fill in all information 4. Save 5. Reopen the user 6. Add extra data |
| Admin – Add house\*\*\* | Log in to the admin panel   1. Click “houses” 2. Click “add house” 3. Add all information    1. Including real estate documents and status 4. Save the house 5. Check all data |
| Admin – Add slideshow\*\*\* | Log in to the admin panel   1. Click “slideshow” 2. Click “add picture to slideshow” |
| Admin – Add Address\*\*\* | Log in to the admin panel   1. Click “address” 2. Click “add address” 3. Fill out all information |

\*Must be performed on every page

\*\*Public contact and sales platform contact

\*\*\*Panels in the admin panel are available from several pages.   
These functionalities are linked through Python Django objects.

## Resources

* Development version available (either on server or on localhost)
* Client / browser

## Tools

Some of these tests must be rechecked multiple times through development. It’s possible to automate the actions by using browser automation software.

* Selenium Browser Automation

## Documentation

See appendix: Exploratory Testing – Activities V1.0.xslx

## Code coverage

All parts have been tested manually. Basic functions for navigation checks have been automated in Selenium. Development changed the site ID’s thoroughly not all tests can be run anymore without adaptation to the last version of the platform.

## Results

|  |  |  |
| --- | --- | --- |
| Functionality | Results | When |
| Navigation | Passed – Automated | 4th sprint |
| Page content | Passed | 7th sprint |
| Keyword search | Passed – Automated | 6th sprint |
| Advanced search | Passed – Automated | 6th sprint |
| Language popup | Passed | 8th sprint |
| Contact forms | Passed – Automated | 5th sprint |
| Login | Passed | 7th sprint |
| Admin – Add users | Passed | 8th sprint |
| Admin – Add house | Passed | 8th sprint |
| Admin – Add slideshow | Passed | 8th sprint |
| Admin – Add Address\*\*\* | Passed | 8th sprint |

# Usability - & UAC tests

## Activities

Usability tests have been executed to test key functionalities. These tests were performed by first year applied informatics students. (Peer tutoring). Due to timing in the development the admin panel could not be submitted to usability tests.

|  |  |
| --- | --- |
| Functionality | Activities |
| Houses overview | 1. Open site 2. Click “buy” |
| Keyword search | Test input/ output   1. Fill in test data 2. Click search   Check if search data is shown correctly |
| Advanced search | Test criteria selection/ output   1. Fill in test criteria to match 1 certain house 2. Click search 3. Check if search data is shown according to criteria |
| Contact forms | 1. Open site 2. Click “Contact” 3. Fill in the necessary information    1. Fill in e-mail address    2. Fill in the message    3. Optional: Choose an attachment 4. Pass the CAPTCHA 5. Click “submit” |

## Resources

* Development version available (either on server or on localhost)
* Client / browser

## Tools

These tests have been automated by first year applied informatics students. (Peer tutoring).

* Selenium Browser Automation

## Code coverage

Usability tests are limited to the 4 scenarios listed in 3.1 Activities.

User Acceptance Tests must be run by real system users. The same scenarios used for usability tests will be provided to Ms. Andriesen after the demonstration. \*

\*Applies only if our application is selected for go-live.

## Results

|  |  |  |
| --- | --- | --- |
| Functionality | Results | When |
| House overview | Passed – Automated | 4th sprint |
| Search | Passed – Automated | 4th sprint |
| Contact forms | Passed – Automated | 4th sprint |

# Security tests

## Activities

|  |  |
| --- | --- |
| Functionality | Activities |
| Login | 1. Click login 2. Fill in correct / incorrect login information 3. Check redirection / error handling |
| Server configuration | * Firewall setup - Port forwarding / allowed services |
| CSRF protection | * Control of CSRF token implementation |
| HTTPS\* | * Control of HTTPS Certificates |

\* Can only be applied to deployment server.

## Resources

* Access to deployment server
* Client / browser

## Tools

* PuTTy – SSH client
* HTTPS Certificate

## Code coverage

Security tests are limited to the 4 scenarios listed in 4.1 Activities.

## Results

|  |  |  |
| --- | --- | --- |
| Functionality | Results | When |
| Login | Passed  Login is handled by Django – Python framework. This means all security protocols are included by the framework. The password is protected by a salted has in the database. | 4th sprint |
| Server configuration | Passed  Allowed services: SSH / HTTP / HTTPS  Allowed ports: 22 / 80 / 443 | 8th sprint |
| CSRF protection | Passed  Django-Python framework has built-in support for CSRF tokens. All data sent on POST is safe from being tampered with. | 5th sprint |
| HTTPS\* | Failed  Configuration is done. Only the required certificates and the deployment server are required to add this security measure. | 8th sprint |

# Multi-platform - & multi-browser tests

## Activities

|  |  |
| --- | --- |
| Functionality | Activities |
| Google Chrome | * Run exploratory tests in Google Chrome |
| Mozilla Firefox | * Run exploratory tests in Mozilla Firefox |
| Microsoft Edge | * Run exploratory tests in Microsoft Edge |
| Apple Safari | * Run exploratory tests in Apple Safari |
| Smartphone\* | Test rescale ability on most popular smartphones  (iPhone / Android phones) |
| Tablet\* | Test rescale ability on most popular tablets  (iPad / Android tablets) |
| Desktop\* | Test rescale ability on most popular screen resolutions.  (1366x768 / 1600x1200 / 1920x1080) |

\*Tested on most common resolutions/platforms.

## Resources

* Development version available (either on server or on localhost)
* Clients
  + Desktop / laptop with screen in different resolutions
  + Android phone / iPhone
  + Android tablet / iPad

## Tools

* Google Chrome
* Mozilla Firefox
* Apple Safari
* Microsoft Edge

## Code coverage

Google Chrome / Mozilla Firefox / Microsoft Edge have been thoroughly tested on desktop / laptop in different screen resolutions.

Multi-platform tests have been run according to availability of the necessary platforms. Since access to iOS / Apple was not widely available only the public site has been tested in Safari / Mac OS. / iOS.

## Results

|  |  |  |
| --- | --- | --- |
| Functionality | Results | When |
| Google Chrome | Passed | 4th – 8th sprint |
| Mozilla Firefox | Passed | 4th – 8th sprint |
| Microsoft Edge | Passed | 8th sprint |
| Apple Safari | Failed – Not enough time spent testing | 8th sprint |
| Multi-platform test | Passed – Scale ability tested on most common devices | 8th sprint |

# Stress/load tests

## Activities

|  |  |
| --- | --- |
| Functionality | Activities |
| Multiple users – load | Run the online tool to generate 25 virtual users |

## Resources

* Deployment server
* Client / browser

## Tools

* Loadimpact.com\*

If our application is selected, a more advanced tool will be needed to check the deployment /hosting server.

\*Free load/stress online test. It creates 25 virtual users.

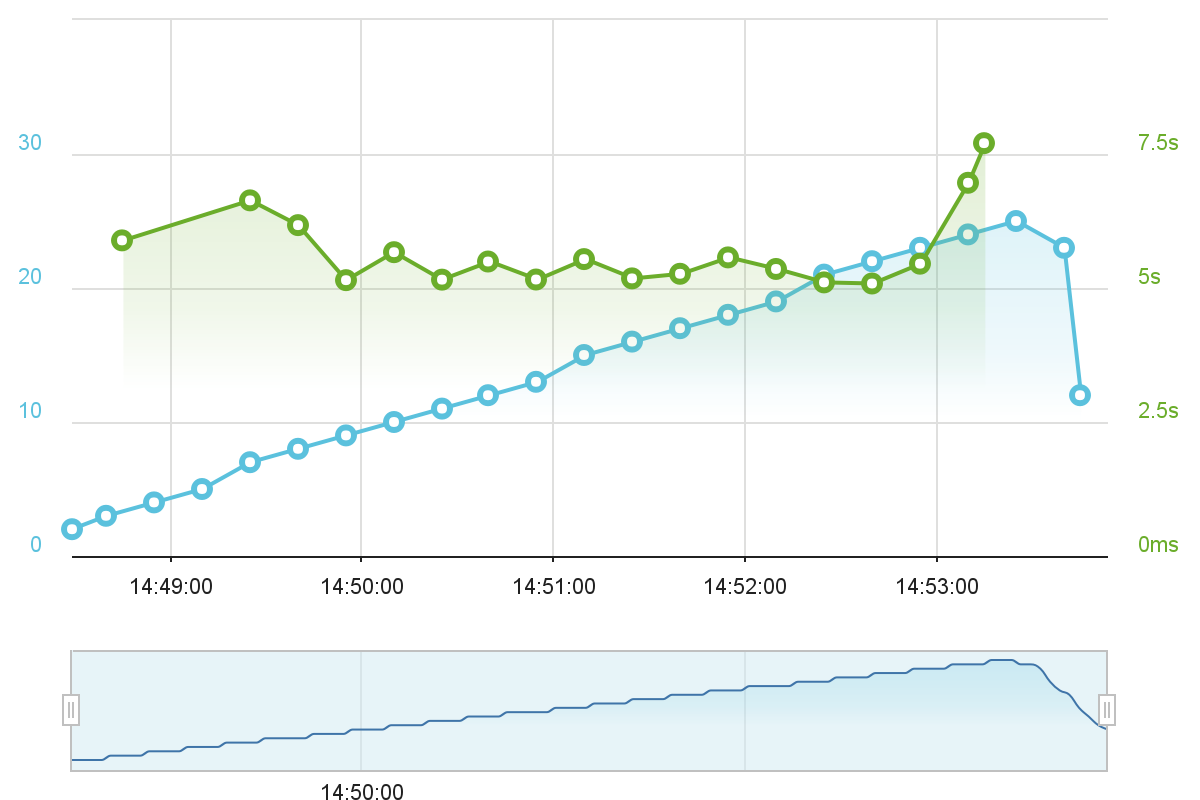
## Code coverage

Not applicable.

The server was submitted to the online stress test.

## Results

|  |  |  |
| --- | --- | --- |
| Functionality | Results | When |
| Multiple users – load | Passed | 8th sprint |



The blue line is an indication as to how the load time increases if more users load the website. The link between amount of users and load time is direct proportional.

We tested our platform on a Digital Ocean environment. These results should be an indication for the chosen hosting service.

# Conclusion

Testing is essential to the development of an application. Though it is not always easy to do, neither is documenting it or checking whether a test is passed or failed. All of these things are essential to developing a good platform/website/application.

During 9 weeks of development our application was tested a lot. Most of these tests were exploratory tests run by the development team to check if a new functionality was working as planned. During the last week of development stress tests, multi-platform tests, multi-browser tests, security tests were added to the portfolio of test results.

Our platform managed to pass almost all tests regarding functionalities of the platform. Most of these were run multiple times to pass. All functionalities were tested several times by different team members over different sprints to make sure the application performed as required.

Security and load tests were added to the test portfolio during the last sprint. These tests are in reality only useful if run on the deployment environment, which was not available at the time of writing. As such these results are an indication of the “outcome to-be” for the deployment server.

# Appendix

* Exploratory Test – Activities V1.0.xlsx
* Selenium tests:
  + Navigation
  + Advanced search
  + Keyword search
  + Login
  + House overview
  + House details
  + Contact form
* Acceptance Test Scenarios (dutch)
  + Hashtags data (Written results)
  + Pompernikkels data (Written results)