**Project Sigma**

**Test Plan**

**Version: 0.5**

**Version control:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Status | Authors | Mentions |
| 0.1 | 29.10.2015 | *Template* | *Juho Juvani* | *First structure of the document* |
| 0.2 | 31.10.2015 | *Draft* | *Juho Juvani* | *Added content* |
| 0.3 | 1.11.2015 | *Draft* | *Juho Juvani* |  |
| 0.4 | 1.11.2015 | *Draft* | *Juho Juvani* | *Added Resources* |
| 0.5 | 2.11.2015 | *Draft* | *Juho Juvani* | *2.4 memory-leak correction – requires schedule to be added and technical requirements for server, and approval for baseline v 1.0* |

Project Manager: Imtiaz Ahmed 0444588

Requirement Analyst: Kuchimanchi Lakshmi Prasanna 0433913

Softaware Designer: Joonas Maksimainen 0372184

Developer: Vitezslav Kriz 0457494

Designer and Developer: Edard Telezhnikov 0460339

Software Tester: Juho Juvani 044472

Table of content

[1. OVERVIEW 3](#_Toc434207338)

[1.1 Purpose 3](#_Toc434207339)

[1.2 Scope 3](#_Toc434207340)

[2. REQUIREMENTS FOR TEST 3](#_Toc434207341)

[2.1 Functional requirements 3](#_Toc434207342)

[2.2 Non-functional requirements 5](#_Toc434207343)

[2.3 Performance testing 8](#_Toc434207344)

[2.4 Reliability testing 8](#_Toc434207345)

[3. RISKS AND TEST PRIORITIES 8](#_Toc434207346)

[4. TEST STRATEGY 9](#_Toc434207347)

[5. RESOURCES 10](#_Toc434207348)

[5.1 Human resource 10](#_Toc434207349)

[7. DELIVERABLES 10](#_Toc434207350)

# 1. OVERVIEW

## 1.1 Purpose

The purpose of this document is to define the following:

* The scope of the testing, areas of focus and objectives
* The responsibilities in the testing
* The testing methods and type of tests
* The test deliverables

## 1.2 Scope

This document details the testing that will be performed by the project team for project Sigma. The documentation defines the requirements and objectives of the testing, by documenting:

* What will be tested;
* How will tests be performed;
* What are the objectives of the tests;
* What is required for the tests to be performed;
* When the tests will be performed

# 2. REQUIREMENTS FOR TEST

In order for tests to be executed, the following requirements must be met:

1. Server with functional website running for testing.

2. Accessible either in university, online, or locally in closed testing environment.

Final version tests must be ran in online environment to validate functionality as detailed in requirement documentation. If possible, outside party should be used to do final use case testing so that genuine reaction and timing can be achieved.

## 2.1 Functional requirements

The project has essential functional requirements that must be validated through testing, as per the requirement documentation. In order for the project to be accepted, the essential functions listed in Table 1 must be validated and tested for functionality as required features.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Functionality** | **Description** | **Type** |
| FR01 | Login | User can login. | Essential |
| FR02 | Logout | Users can logout. | Essential |
| FR03 | Register | User can sign up. | Essential |
| FR04 | Delete account | Users can delete their own account. | Essential |
| FR05 | Edit account info | User can edit information of its account. (name, email, password) | Essential |
| FR06 | Own organization | User is owner of organization, recruiter, student, moderator or admin. | Essential |
| FR07 | Edit other accounts | Moderator can edit student-, recruiter- and organization –accounts. | Essential |
| FR08 | Edit user rights | Administrator can add and remove other users and moderators. | Essential |
| FR09 | Administrator role extends moderator | Administrator role includes moderator capabilities. | Essential |
| FR10 | Owner creates accounts | Owner is the creator of the organization in the system and can add recruiters to its organization. | Essential |
| FR11 | Approve organization | Moderator can approve the creating of organization account. | Essential |
| FR12 | Employment | Owners of organizations and recruiters can belong to multiple organizations. | Essential |
| FR13 | Post job listing | Owners of organizations and recruiters can post job listings. | Essential |
| FR14 | Invite to interview | Owners of organizations and recruiters can invite students to interview. | Essential |
| FR15 | Student search | Owners of organizations and recruiters can search for students by skills etc. | Essential |
| FR16 | Search parameter selection | User can make different searches by different parameters. | Essential |
| FR17 | Reply to invitation | Student can reply to the invite positively and negatively. | Essential |
| FR18 | Document sharing | Student can add and show actual assignment thesis and reports. | Essential |
| FR19 | Verify users | Moderator can verify other users. | Desirable |
| FR20 | Comment | Student can add explanation to invite reply. | Desirable |
| FR21 | Edit profile | User can edit other information of its account. (photo, bio, content information) | Desirable |
| FR22 | Job feed | Student can view latest job listings. | Desirable |

Table 1 Functional requirements

## 2.2 Non-functional requirements

Additionally, Table 2 lists the non-functional requirements presented in the requirements documentation. These non-functional requirements will not be the focus of testing, but certain features must be validated through use-case testing and creating different scenarios for a tester to take in order to validate the requirements. These include:

* Usability testing with the user interface (Errors, intuitiveness of the layout and structuring of the site)
* Reliability (No empty pages, or errors to be faced in normal use)
* Performance and timing, measured during use case testing
* Security testing

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **Description** |
| NFR1 | User interface should be straightforward and simple to use | Minimal amount of clicks |
| NFR2 | User interface should be intuitive | Logic of using the site should be clear in short time after looking over the first page |
| NFR3 | Site should have clear and simple instructions on how to use it (Both search and registration) | Easy instructions on how to register as a user and how to search offers with minimal confusion |
| NFR4 | Reliability | There should not be empty pages or errors appearing on the site. |
| NRF5 | Predictability | Search functions should provide predictable results:   * Specific job offers according to keywords * Student search should return all fitting skills and talent from students according to search |
| NFR6 | Accuracy of information | Both job offers and user information should be up to date.  Job offers should be timed to be visible or have their status changed at the end of recruitment dates.  If student graduates, information should be hidden or status changed within reasonable time. |
| NRF7 | Performance | Response delay 1-3 seconds, pending on traffic and users connection. |
| NRF8 | Availability | Site uptime should be 24/7 |
| NRF9 | Supportability | Site should be easily managed by administrators and moderators with minimal effort.   * Post edit * Post removal * Profile control (Hide / Show profile) * Edit Profile / Company site and information |
| NRF10 | Database | Data must be easily controlled and privacy of user information guaranteed. |
| NRF11 | Language (English) | Site should be available in English |
| NRF12 | Server (Linux) | Site must be hosted and held on a Linux server provided by the LUT |
| NRF13 | Security | SIGMA is concerned for the individual privacy issues, it has special features to allow and restrict information when needed, it also have many protocols like web security tools and standard security technology like SSL to overcome security issues related with passwords and file uploads. There will be a check on following issues   1. Duplication of files 2. Authorization 3. Authenticity 4. Validity of data |
| NRF14 | Capabilities | The will be error free edition of website, where the typos and grammatical errors will be identified and corrected automatically, allowing the users a error free copy of the contents. In addition to this SIGMA also provides a user friendly system which can be easily accessed by a naive user. |
| NRF15 | Usability | The main aspect of usability is availability and accessibility of the website. If the website is not reachable by the students and the recruiters than that website is totally useless.  **Server time:** The SIGMA portal is able to easily access website without any errors in loading the page. The contents will be loaded all at once.  **Broken Links:** The SIGMA will not have any broken links  **Clarity:** The information in the SIGMA will be consistent with easy navigational architecture.  **Credibility:** The SIGMA will be trustworthy to acquire a dream job.  **Relevant information:** The contents in the SIGMA will be relevant without any ambiguity.  **View and Appearance:** The website gives a professional environment with easy navigation and explicit display. |
| NRF16 | Design Constraints | The design content will have the first view with simple view of a window with all necessary functionalities and with design flexibility. The contents can be viewed on personal computers, laptops and tablets. |
| NR17 | Mean time between failures | MTFB will mostly less than 24hours during working days, and during weekends based on availability of technical support. |

Table 2 Non-functional requirements

## 2.3 Performance testing

For performance testing the following are points of focus:

* Response time measured and recorded in use case testing
* Resource usage
* Different browsers (Mozilla, Explorer, Opera, Chrome, Edge, Safari, Others(?))
* Different configurations or constraints

## 2.4 Reliability testing

As the project uses Python for coding, memory-leaking is not considered to be a threat because of its functionality for preventing it. For reliability testing the following are to be focused:

* Resistance to failure and run-time errors (Timeouts in data search, errors, warning messages and recovery from error)
* Code integrity and structure (During unit testing and development)
* Resource usage

# 3. RISKS AND TEST PRIORITIES

Major risks must be identified in order for the testing to be successful, so that potential serious exposure can be limited to internal discovery. List of risks will be iterated over the course of development and testing, based on the required functional features and potential constraints of the project. The risks will be assessed by effects as follows:

|  |  |  |
| --- | --- | --- |
| **Risk** | **Risk factor** | **Risk description** |
| Any found functionality that can pose a risk in case of malfunction or error | Risks will be categorized with following factors:  **H** – High risk, must be fixed before release. Poses serious conflict and issue with requirements.  **M** – Medium risk, can be accepted if it does not break main functionality or pose a risk to privacy, usability or integrity of the system  **L –** Low risk, acceptable. Little to no effect on the actual functionality of the system or users. | More detailed description of the risk and justification. |

Table 3 Risk assessment

# 4. TEST STRATEGY

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Type** | **Objectives** | **Completion** | **Technique** |
| **Function testing** | * That the features meet the defined requirements * User interfaces work correctly * Data is displayed correctly * Errors are handled correctly | * All planned tests have been executed * All identified defects in the testing are fixed | * Executing each use case and use case flow * Using valid and invalid data |
| **User interface testing** | To verify that:   * Navigation and layout meets the requirements of usability and function * User interface appears identical and functional on different platforms (browsers) | Each page is verified to function according to requirements, with multiple platforms | Test all windows with different users, different data and different browsers.  Lastly test scalability of the windows. |
| **Data and Database Integrity** | * Ensure that database access and retrieval of data functions properly * Ensure that no data corruption is apparent * Ensure that no data entries are corrupted, mixed or mistaken during multiple user workload | All database access functions as intended and cannot be accessed outside the designed processes | * Invoke different requests or access to database, with both valid and invalid data * Inspect that data has been entered as intended, or that correct data was retrieved from right table and row |
| **Performance testing** | * Ensure that basic functions and page loading works under load of multiple users * Verify performance under worst expected workload | * Succesful completion of entries without errors under normal load * Succesful completion of entries without errors under heavy load | * Test with multiple users manually * Modify datafiles to increase number of transactions * Run a scripted entry of multiple connections to the page |

Table 4 Test types

# 5. RESOURCES

The main resource for Sigma project is the project team.

## 5.1 Human resource

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| **Project Manager (Imtiaz)** | Management oversight  Acquires appropriate resources  Management reporting |
| **Requirement Analyst (Prasanna)** | Confirms that requirements are validated as documented  Makes changes to requirements upon requests |
| **Designer (Joonas)** | Assists in generating testing documentation when needed  Assists in executing tests  Makes design changes upon request  Approves testing of design features |
| **Developers (Eduard and Vita)** | Execute unit tests   * Log results of tests * Document changes and change requests |
| **Tester (Juho)** | Generates Test Plan  Generates Test Cases  Evaluates progress of testing and its effect  Writes Test Evaluation Report  Executes the use case tests   * Logs results of tests * Document change requests |

Table 5 Human resource listing

# 7. DELIVERABLES

The Testing phase will aim to produce the following deliverables:

1. Test Plan

2. Test Cases (For each use case)

3. Test Results

4. Test Evaluation Report