Team SuperWoman

SuperWoman
Test Plan

Version 1.1

Revision History

Date	Version	Description	Author
02.05.2018	1.0	First Version	Team SuperWoman
20.06.2018	1.1	Reviewed Document	Team SuperWoman

Table of Contents

Test Plan

1. Introduction

1.1 Purpose

The purpose of the Iteration Test Plan is to gather all of the information necessary to plan and control the test effort for a given iteration. It describes the approach to testing the software, and is the top-level plan generated and used by managers to direct the test effort.

This *Test Plan* for the SuperWoman project supports the following objectives:

- Outlines the testing approach that will be used
- Identifies the items that should be targeted by the tests
- Identifies the required resources.

1.2 Scope

This document addresses the following types and levels of testing:

- Unit-Tests
- Functionality Testing.

1.3 Intended Audience

This document addresses the project members of SuperWoman and everybody who is interested in the SuperWoman-Project.

1.4 Document Terminology and Acronyms

n/a

1.5 References

- GitHub
- Blog

2. Evaluation Mission and Test Motivation

2.1 Background

By testing our project, we make sure that all changes to the sourcecode do not break the functionality. Furthermore, by integrating the test process in our deployment process, we make sure that only working versions of our project get deployed. This renders possible that the web application is always available.

2.2 Evaluation Mission

Our mission for the evaluation effort of the current iteration is following:

- Find as many bugs as possible as early as possible
- Validation of our sourcecode.

2.3 Test Motivators

Using tests makes possible to validate the correctness of the methods and the functionalities in our sourcecode. Therewith, bugs can be found as early as possible and major effects can be prevented.

3. Target Test Items

The listing below identifies those test items—software, hardware, and supporting product elements —that have been identified as targets for testing. This list represents what items will be tested:

- Navigation within the website
- UI
- Functionality Testing.

4. Outline of Planned Tests

4.1 Outline of Test Inclusions

- JUnit-Testing
- Cucumber UI-Testing.

4.2 Outline of Other Candidates for Potential Inclusion

n/a

4.3 Outline of Test Exclusions

n/a

5. Test Approach

5.1 Initial Test-Idea Catalogs and Other Reference Sources

n/a

5.2 Testing Techniques and Types

5.2.1 Data and Database Integrity Testing

We are still working on implementing a database, thus we are unable to test this section presently.

5.2.2 Function Testing

Technique Objective:	Testing if the UI fits the expectations.
Technique:	We try to cover all use cases within one test by executing them one after another.
Oracles:	We are relying on the outcome of our testing tool.
Required Tools:	Jenkins
Success Criteria:	The tests pass without error messages.
Special Considerations:	-

5.2.3 Business Cycle Testing

n/a

5.2.4 User Interface Testing

Technique Objective:	Testing of the User Interface (test of the usability of our website)
Technique:	Usability Tests: Users should navigate through the website and play the game
Oracles:	The user likes the website. The website is easy to use and the menu navigation is simple and intuitive.
Required Tools:	User, Laptop with internet access
Success Criteria:	Website has a good usability and the users like our game.

5.2.5 Performance Profiling

n/a

5.2.6 Load Testing

n/a

5.2.7 Stress Testing

n/a

5.2.8 Volume Testing

n/a

5.2.9 Security and Access Control Testing

n/a

5.2.10 Failover and Recovery Testing

n/a

5.2.11 Configuration Testing

n/a

5.2.12 Installation Testing

n/a

6. Entry and Exit Criteria

6.1 Test Plan

6.1.1 Test Plan Entry Criteria

The execution of the test plan can begin as soon as the first unit test is written.

6.1.2 Test Plan Exit Criteria

The execution of the test plan is completed when the project is finished.

6.1.3 Suspension and Resumption Criteria

n/a

6.2 Test Cycles

6.2.1 Test Cycle Entry Criteria

A test cycle begins when the build of a new version is initiated.

6.2.2 Test Cycle Exit Criteria

A test cycle ends when the build of a new version fails.

6.2.3 Test Cycle Abnormal Termination

A test cycle ends abnormally if an error occurs during the build.

7. Deliverables

7.1 Test Evaluation Summaries

Test evaluation is done by hand directly after the test has been executed.

7.2 Reporting on Test Coverage

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio
SuperWoman	39,4 %	282	434	716

bd

8. Testing Workflow

Unit tests run automatically in the IDE, we start the functional tests after every commit.

9. Environmental Needs

9.1 Base System Hardware

The following table sets forth the system resources for the test effort presented in this *Test Plan*.

System Resources		
Resource	**Quantity**	**Name and Type**
Database Server		
Network or Subnet		Internet
Server Name		Postgres Server
Database Name		PostgreSQL
Client Test PCs		
Include special configuration requirements		Özen, Lea, Isabella
Test Development PCs		Özen, Lea, Isabella

9.2 Base Software Elements in the Test Environment

The following base software elements are required in the test environment for this *Test Plan*.

Software Element Name	Version	Type and Other Notes
Windows	Win 8 or higher	Operating System
Google Chrome		Internet Browser
Eclipse		IDE
JUnit	latest	
Selenium	latest	

9.3 Productivity and Support Tools

The following tools will be employed to support the test process for this *Test Plan*.

Tool Category or Type	Tool Brand Name	Vendor or In-house	Version
Project Management	JIRA	Atlassian	7.0

9.4 Test Environment Configurations

The following Test Environment Configurations needs to be provided and supported for this project.

Configuration Name	Description	Implemented in Physical Configuration

n/a

10. Responsibilities, Staffing, and Training Needs

10.1 People and Roles

This table shows the staffing assumptions for the test effort.

- Requirement Specifier: Özen, Lea & Isabella
- Designer: Özen
- Implementer: Özen, Lea & Isabella
- Tester: Özen, Lea & Isabella

- Project Manager [Deadline Manager]: Isabella
- Implementer [Code Cleaner]: Lea

11. Iteration Milestones

[Identify the key schedule milestones that set the context for the Testing effort. Avoid repeating too much detail that is documented elsewhere in plans that address the entire project.]

Milestone	Planned Start Date	Actual Start Date	Planned End Date	Actual End Date
>20% Test Coverage	09.05.2018	16.05.2018	13.06.2018	20.06.2018
Have functional tests	09.11.2017	12.11.2017	13.11.2017	13.11.2017
Have Junit tests	20.04.2018	02.05.2018	10.06.2018	20.06.2018
Have end user test	13.06.2018	13.06.2018	19.06.2018	19.06.2018

12. Risks, Dependencies, Assumptions, and Constraints

The risk for our project are outlined in our Risk Management Table.

There are no additional risks for testing.

13. Metrics

We used the online tools Codacy, SonarQube and CodeClimate.
The first metric that has been tested is code duplication. Duplicated code can lead to software that is hard to understand and difficult to change.
The second one we tested is cognitive complexity. Cognitive Complexity is a measure of how difficult a unit of code is to intuitively understand. Unlike Cyclomatic Complexity, which determines how difficult your code will be to test, Cognitive Complexity tells you how difficult your code will be to read and comprehend.