|  |
| --- |
| Test Plan |
| Webshop |

|  |
| --- |
| by Blank Team  Tóth Noémi, Farkas Balázs, Barkóczi Martin, Dombi Tibor, Szegedi József |

**GamerzWebshop Project:**

* **Course name:** Application Development Project Lab
* **Instructor:** Tilki Csaba
* **Time:** Monday 13:40-15:10
* **Room:** C-108
* **Project description:** Webshop development in PHP
* **Team name:** Blank Team
* Members:
  + Tóth Noémi
  + Farkas Balázs
  + Barkóczi Martin
  + Dombi Tibor
  + Szegedi József

**Document Change Log:**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Change Date | By | Description |
| 0.1 | 2019-11-15 | Dombi Tibor | Create test plan |
| 0.2 | 2019-11-22 | Dombi Tibor | Format and Style |
| 1.0 | 2019-11-22 | Dombi Tibor | Revision |

Contents

[1 Introduction 3](#_Toc25325428)

[1.1 Scope 3](#_Toc25325429)

[1.1.1 In Scope 3](#_Toc25325430)

[1.1.2 Out of Scope 4](#_Toc25325431)

[1.2 Quality Objective 4](#_Toc25325432)

[1.3 Roles and Responsibilities 4](#_Toc25325433)

[2 Test Methodology 4](#_Toc25325434)

[2.1 Overview 4](#_Toc25325435)

[2.2 Test Levels 4](#_Toc25325436)

[2.2.1 Functionality testing: 5](#_Toc25325437)

[2.2.2 Usability Testing 5](#_Toc25325438)

[2.2.3 Interface Testing 5](#_Toc25325439)

[2.2.4 Database Testing 6](#_Toc25325440)

[2.2.5 Compatibility Testing 6](#_Toc25325441)

[2.3 Bug Triage 6](#_Toc25325442)

[2.4 Test Completeness 7](#_Toc25325443)

[3 Test Deliverables 7](#_Toc25325444)

[4 Resource & Environment Needs 7](#_Toc25325445)

[4.1 Testing Tools 7](#_Toc25325446)

[4.2 Test Environment 8](#_Toc25325447)

[5 Terms/Acronyms 8](#_Toc25325448)

# Introduction

The Test Plan is designed to prescribe the scope, approach, resources, and schedule of all testing activities of the project.

The plan identify the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

## Scope

### In Scope

All the feature of the website which were defined in software requirement specs are need to tested

|  |  |  |  |
| --- | --- | --- | --- |
| Module Name | As Guest | As User | As Admin |
| Profile | Register  Log In | Log Out  Check own profile data  Edit own profile | Log Out  Check own profile data  Check the profile data of any user  Edit the profile data of any user |
| Order | Place an order on any item | Place an order on any item  Use predefined shipping address | Place an order on any item  See all orders  Edit any order  Delete any order  Mark any order completed |
| Items | See Items  Use filters on shop page  See rating on items  See comments on items | See Items  Use filters on shop page  See rating on items  See comments on items  Rate any item  Add comment to any item  Edit own comment  Delete own comment | See Items  Use filters on shop page  Add Item  Delete Item  Edit Item  See rating on items  See comments on items  Rate any item  Add comment to any item  Edit any comment  Delete any comment |

### Out of Scope

These features are not be tested because they are not included in the software requirement specifications:

* Hardware Interfaces
* API
* Website Security and Performance

## Quality Objective

We must make sure:

* Ensure the Application Under Test conforms to functional and non-functional requirements
* Ensure the AUT meets the quality specifications defined by the client
* Bugs/issues are identified and fixed before go live

## Roles and Responsibilities

Testing will be executed by:

* Dombi Tibor: Test manager, responsible for test documentations
* Tóth Noémi: responsible for UI testing
* Farkas Balázs: responsible for BE testing (database excluded)
* Barkóczi Martin: responsible for DB testing

# Test Methodology

## Overview

Our test methodology will follow the “waterfall method”:

In the waterfall model, software development progress through various phases like Requirements Analysis, Design etc - sequentially.

In this model, the next phase begins only when the earlier phase is completed.

## Test Levels

In the project, the following tests should be conducted.

### Functionality testing:

This is used to check if your product is as per the specifications you intended for it as well as the functional requirements charted out in developmental documentation. Testing Activities includes:

* Test all links in webpages are working correctly and make sure there are no broken links.  
  Links to be checked will include:
  + Outgoing links
  + Internal links
  + Anchor links
  + MailTo links
* Test Forms are working as expected. This will include:
  + Scripting checks on the form are working as expected. (For example: if a user does not fill a mandatory field in a form an error message is shown.)
  + Check default values are being populated
  + Once submitted, the data in the forms is submitted to a live database or is linked to a working email address
  + Forms are optimally formatted for better readability
* Test Cookies are working as expected. Cookie Testing will include:
  + Testing cookies (sessions) are deleted either when cache is cleared or when they reach their expiry.
  + Delete cookies (sessions) and test that login credentials are asked for when you next visit the site.
* Test HTML and CSS to ensure that search engines can check the site easily. This will include:
  + Checking for Syntax Errors
  + Readable Color Schemas
  + Standard Compliance. Ensure standards are followed.
* Test negative scenarios, such that when a user executes an unexpected step, appropriate error message or help is shown in your web application.

### Usability Testing

Test the site Navigation:

* Menus, buttons or Links to different pages on your site should be easily visible and consistent on all webpages

Test the Content:

* Content should be legible with no spelling or grammatical errors.
* Images if present should contain an "alt" text

### Interface Testing

Three areas to be tested here are - Application, Web and Database Server

* Application: Test requests are sent correctly to the Database and output at the client side is displayed correctly. Errors if any must be caught by the application and must be only shown to the administrator and not the end user.
* Web Server: Test Web server is handling all application requests without any service denial.
* Database Server: Make sure queries sent to the database give expected results.
* Errors: Test system response when connection between the three layers (Application, Web and Database) cannot be established and appropriate message is shown to the end user.

### Database Testing

Database is one critical component the web application and stress must be laid to test it thoroughly. Testing activities will include:

* Test if any errors are shown while executing queries
* Data Integrity is maintained while creating, updating or deleting data in database.
* Check response time of queries and fine tune them if necessary.
* Test data retrieved from your database is shown accurately in your web application

### Compatibility Testing

Compatibility tests ensures that your web application displays correctly across different devices. This includes:

* Browser Compatibility Test: Same website in different browsers will display differently. We need to test if the web application is being displayed correctly across browsers, JavaScript, AJAX and authentication is working fine. It is also mandatory to check for Mobile Browser Compatibility.
* The rendering of web elements like buttons, text fields etc. changes with change in Operating System. We must make sure the website works fine for various combination of Operating systems such as Windows, Linux, Mac and Browsers such as Firefox, Internet Explorer, Safari etc.

## Bug Triage

The goal is to:

* Define the type of resolution for each bug
* Prioritize bugs and determine a schedule for all “To Be Fixed Bugs’.

Priority can be:

* Important – Issues that threatens the security or the integrity of the application
* Urgent – Issues or errors that can render the application unusable
* Normal – Issues that can negatively affect user experience (slow process, unexpected redirections, etc.)
* Below normal – Other issues, like misspelling, mispositioning of elements, design problems, etc.

**Example Bug report:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | App version | Description | Module | Attachments | Priority |
| 2019-11-22 13:47 | 0.1.3 | Unresponsive registration form on IE11 | UI | example/link/to/image.png  example/link/to/file.log | Urgent |
| 2019-11-22 14:10 | 0.1.3 | SQL injection possible on search field | UI, Items | example/link/to/image.png  example/link/to/file.log | IMPORTANT |

On the above example the SQL injection issue has higher priority, subsequently it needs to be solved before anything else.

## Test Completeness

Criterias to check Test Completeness are

* 80% test coverage
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release

# Test Deliverables

The following documents are needed to be delivered during test lifecycle:

* Test Plan
* Test Cases
* Bug Reports

# Resource & Environment Needs

## Testing Tools

The following tool will be used for testing: (all included in *“.debug”* folder)

* Functional testing: Selenium on C#
* Usability testing: Any screenshot tool
* Interface testing: Selenium on C#
* Database testing: various PHP scripts
* Compatibility testing: Selenium on C#

## Test Environment

The person executing tests will need the following software enviroment (or newer versions):

* PHP 7.2
* Apache 2.4.0
* MySQL 8.0
  + MariaDB 10.1.38
* .Net Framework 3.0

# Terms/Acronyms

Terms and acronyms used in the project

|  |  |
| --- | --- |
| TERM/ACRONYM | DEFINITION |
| API | Application Program Interface |
| AUT | Application Under Test |
| BE | Back end, during development mostly used to refer to PHP codes |
| DB | Database |
| FE | Front end, during development mostly used to refer to the HTML and/or CSS code |
| JS | JavaScript |