Test Plan

Navigator

Version:2

2019-03-03

1.1. Contents

2. INTRODUCTION

2.1. Purpose

This document does not provide detailed test cases with steps and expected results. The document's purpose is to ensure that reader understands the scope of the testing, the specific scenarios required to appropriately cover testing of the change and any other impacted existing code.
The purpose of this document is to enable the testers to make objective assessments regarding the degree of conformance of the system to stated requirements and specifications.
In addition, testing should validate that the system being developed is what the user needs. In essence, validation is performed to ensure that we are building the right system. Apart from helping make decisions, the information from software testing helps with risk management. These will be produced and reviewed following approval of this plan.

2.2. Audience

The intended audience for this document is test leaders, project leaders, members of the team, and anyone else who is or will be involved in testing within the project.

3. SYSTEM UNDER TEST

3.1. Description

Navigator.ba is the most popular city guide that provides data about streets, places and events in Sarajevo. Database contains over 4000 places and 1300 streets as well as monthly schedules of local theaters, cinemas and art galleries with the possibility to book your tickets.

Navigator.ba is available on web and mobile including native apps for iPhone/iPad and Android devices.

Users can contribute to development of Navigator., create places from their surroundings, fulfill data for existing places, share content and suggest enhancements that would like to see in near future.

3.2. Purpose

The purpose of Navigator as a city guide is to provide data about streets, places and events in Sarajevo.

3.3. Audience

This application is for everyone who wants to find some place in Sarajevo or Bosnia and Herzegovina.

4. TEST STRATEGY

The approach to the testing activities are described in this section along with the identified test scope, and which test areas and activities will be used during which cycle and who will be part of the testing.

The test strategy for testing is to work according to an Waterfall methodology, because Navigator.ba is completed product.

4.1. Test Areas

Testing will be divided into the following areas:

- Exploratory testing
- Smoke Test
- Regression testing

The implementation will be tested in Google Chrome and Mozilla Firefox as web application.

4.1.1.Out of Scope

The following areas will not be tested :

- Areas/functionality that haven't been delivered.
- Backup of servers or other infrastructure related areas
- Functionality inside integrating systems will not be tested
- Testing on another browser except Google Chrome
- Testing on Mobile Devices
- Performance Test

4.2. Time Planning

Start date	Sync Up	End date	Demo
06.02.2019	25.02.2019	06.03.2019	07.03.2019

4.3. Environment

The test activities will be maintained and provided on production environment.

4.3.1 Operative system and Browsers

Google Chrome is used as preferred Browser, and application is also tested in Mozilla Firefox. Windows and Linux as operative systems. Testing will be done on Windows 10 and Linux/Ubuntu 16.04.

4.3.2 Software testing tools

In first cycle of Manual testing (Smoke, Exploratory, Regression) Google Sheets are used for management of test cases and test activities. And later we used the Lean Testing tool.

Software used for automation testing: Selenium, RSpec, Ruby, Watir/Capybara.

Software	**Description**
Google Sheets	*Google Sheets* are used for management of test cases and test activities for Manual testing.
Lean Testing tool	*Lean Testing* is a free complete test management solution. It's hosted bug tracker is designed for highly productive software development teams, with standardized bug reports, internal/external user management, bug reporting templates that force best practices and standardized bug reports.
Selenium	*Selenium* is a portable framework for testing web applications.
RSpec	*RSpec* is a 'Domain Specific Language' (DSL) testing tool written in Ruby to test Ruby code.
Ruby	*Ruby* is a dynamic, interpreted, reflective, object-oriented, general-purpose programming language.
Watir/Capybara	*Watir and Capybara* are an open-source families of Ruby libraries for automating web browsers.
Trello	*Trello* is the flexible, and visual way to manage and organize projects.

4.4. Test deliverables

Aftes testing process is finished we expect to deliver an application that should be developed according to the requirements. All bugs are successfully reported and some improvements are suggested.

Test deliverables for this application: **Test Cases Documents, Test Plan, Test Scripts, Test results/reports, Test automation report.**

4.5. Risks and proposed solutions

Risk ID	Risk description	Priority
1	The Testing environment may not be setup properly	High
2	There is no enough human resources to finish project at deadline	Medium
3	No communication in the team	Low
4	No advanced technology available or the existing technology is in initial stages	High

4.6. Roles and Responsibilities

Role	**Description**	**Name**
Mentor	Responsible for the test coordination and for Delivery Management. Mentor should also provide advice, guidance, and feedback; share their experience and expertise as appropriate; act as a sounding board for ideas and action plans.	**Amina Alijagic**
Test Engineer	Responsible for test preparation and test reporting, creation of test case scenarios for all areas and tenants together with test execution.	**Medzida Mustafic**
Talent Manager	Present on demo presentation to give feedback and help make an overview of the entire testing process and progress of a test engineer.	**Nina Cengic**
Project Coordinator	Coordinating project schedules, resources, equipment and information. Liaising with clients to identify and define project requirements, scope and objectives. Ensuring that clients' needs are met as the project evolves	**Adna Karkelja**

5 TEST ACTIVITIES

5.1. Writing and executing test cases

Test cases are written and executed in Google sheets. When writing test cases, we will use the user case approach, which means that we will test using scenarios. Test scenarios will also be used to create as comprehensive test cases as possible to suit all inputs and process combination. The test cases shall have one or several scenarios steps. Some with expected results and some without, depending on the requirement and the scenario. Test cases are divided into folders such as smoke test, exploratory tests and regression tests. It shall be possible to reuse the test cases between the test activities and the inputs.

A test report will be delivered showing graphs presenting the total number of test cases, total number of run test cases and how many of these that failed (if any) will be showed. Also, the number of found defects will be presented.

5.2. Test Cases

Test cases are created and executed manually for Exploratory and Regression testing. Smoke test is created and executed manually and then automation test script for smoke test is created using tools for automation testing.

Entry Criteria

- Testing resources have been identified and are available for testing.
- Test scope, approach, responsibilities and timeline are confirmed with and communicated to all relevant participants.
- Tenant has been successfully configured and delivered
- Smoke test has been performed by the implementers and has passed
- All testers have access to the tenant with correct security roles, login, and password
- Test tools are available for use during testing
- Access to the page is available.
- New testers have been educated in specific test tools to start testing

Exit Criteria

- 100% test coverage

- Successful completion of planned scenarios with actual results is documented within specific test tools
- No unresolved and re-tested defects with High level priority
- No unresolved and re-tested defects with Medium level priority
- No unresolved and re-tested defects with priority level Low.
- All remaining defects (not closed or rejected) are documented.

5.3 Automation Testing

For this project type of software testing that will be automated is Smoke test, using a Selenium, Rspec, Ruby, Capybara. Automation Scripts are created and executed during this phase by test engineer.

6 DEFECT MANAGEMENT

Lean testing tool and Google sheets will be used as the defect reporting tools. If a defect is found during testing, it's important that a reference to the test case is added to increase the traceability but also to find potential issue areas.

6.1. Defect Process

Each user conducting test will report and relay any new issues to the Test Lead. All incidents should be also being reported. Each defect shall contain the following information:

Bug ID:

- Bug ID
- Name
- Reporter
- Submit date

Bug overview:

- Summary
- URL
- Screenshot

Environment:

- Operating system

- Browser

Bug details:

- Steps to reproduce
- Expected result
- Actual result
- Description

Bug tracking:

- Severity
- Assigned to
- Priority

Notes:

- Notes

Everything that could be of interest when troubleshooting should be attached in the defect, i.e. screenshots.

The test manager is responsible for all the defects reported within the given test level and for closing the bugs. Test manager is also responsible to conduct and lead bug reports meetings. During these meetings, it should be decided if the bugs should be fixed or not, review the priority of the bug to make sure the bug fix is done in a correct order.

7 EXPECTED RESULTS

Expected results after test activities are to have an application developed according to the requirements, fix all reported bugs, and apply suggested improvements.

8 OTHER

8.1. Meetings

Weekly meeting -

There will be a status meeting held every week. Test progress, status, hinders and problems as well as defects are the focus of this meeting.. Attendees are tester and mentor. In short, the

people that are doing the testing will be invited to these meetings. The moderator for these meetings is the mentor.

Sync Up -

A monthly meeting to ensure that the project is going forward on IT's behalf. This means that the status, ongoing and upcoming activities, potential risks, actions and new decisions is discussed. Attendees are representatives from IT, such as Project Coordinator, Tester, Mentor.

Demo -

Demo and presentation of finished tasks in last month are presented and discussed.

8.2. Terms and definitions

Term	Definition
Test Case	A set of input values with expected, to verify compliance requirement.
Test Plan	Will detail out specifics of test approach for each area and type of testing.
Test Type	A group of test activities aimed at testing a component or system focused on a specific test objective

Note: This document should be reviewed by the QA Lead, prior to distribution to the wider audience.