|  |
| --- |
|  |
| Teodent Website |
| Test plan |
|  |
| **Andriesei Diana** |
| **Grupa 7** |

|  |
| --- |
| A test plan is a formal document that details the process of testing a software product. It includes the test objectives, requirements, schedule, resources and testing methods, as well as the rules for accepting the results. The purpose of the test plan is to provide a systematic and coherent testing approach to ensure that all critical components of the software product are adequately tested. |

Test plan – Teodent

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 7/6/2023 | Version 1.0 | Andriesei Diana | Drafts |

**Table of content:**

1. **Introduction**
   1. Test Objectives
   2. Functionalities in scope
   3. Functionalities out of scope
2. **Test process**
   1. Test planning
   2. Test analysis
      1. Techniques used for GUI
      2. Test condition for the “Programeaza-te” and “Contact” category
   3. Test design
   4. Test implementation
   5. Test execution
   6. Test closure
   7. Test monitoring and control
3. **Test deliverables**

* Test plan
* Test conditions
* Test cases
* Daily test summary reports
* Traceability matrix
* Test case results
* Bugs report

1. **Introduction:**

The purpose of this plan is to provide an understanding of the testing method used on the <https://teodent.ro/> website, with special attention to the “Contact” and “Programeaza-te” sections. To guarantee an end-to-end flow for the office appointment scheduling and clinic contact processes, the test scope includes GUI and functional testing techniques.

* 1. **Test Objectives**
* The objective of this project is to apply the knowledge and skills acquired in a real-world testing experience.
* The purpose of this project is to gain real-life experience by implementing a real application and demonstrating the ability to apply theoretical knowledge.
* Verify the functionality and ease of use of the website https://teodent.ro/, particularly the “Programeaza-te” and “Contact” sections.
* The objective is to ensure a seamless end-to-end flow for scheduling an appointment at the clinic and contacting the clinic.
* Identify and report any issues or defects encountered during testing.
  1. **Functionalities in scope**
* Successfully obtaining a booking by successfully completing the appointment form.
* The possibility to get in touch with the office by using the contact form.
  1. **Functionalities out of scope**
* Non-functional testing encompasses performance, security, and compatibility testing.
* Testing on specific hardware configurations or devices.
* Testing functionalities that refer to other types of products.
* To date, there is no solution that provides assistance in testing the quality of mobile applications.
* Automated testing.

1. **Test process**
   1. **Test planning:**

* Identification of roles and responsibilities:

|  |  |
| --- | --- |
| **Software Developer** | Andrei Andrei |
| **Product Owner** | Group 7 Manual Testing |
| **Project Manager** | IT Factory |
| **QA Engineer** | Diana Andriesei |
| **Senior QA Engineer** | Vasilica Onuta |

**Entry Criteria:**

Before testing begins, the following entry requirements must be met:

* The application should be up and running and available for testing.
* The necessary test data and environment must be configured.
* Requirements documentation should be accessible for consultation.
* Testing tools and resources should be prepared and available for use.

**Exit Criteria:**

To consider the testing process complete, the following exit requirements must be met:

* All scheduled test cases have been completed.
* Issues and defects were identified, documented and tracked to resolve.
* Created and reviewed test reports and documentation.
* The test objectives set by the project stakeholders were met.
* The deadline has been reached.

**Risks:**

During the test project, the following potential threats were identified:

**Project risks:**

* Resource Constraints: Lack of skilled resources such as developers, testers or subject matter experts could have a negative impact on project progress and quality.
* Scope limitation: Frequent changes to project scope without proper assessment and impact analysis could disrupt project timelines, resources, and deliverables.
* Communication issues: Lack of communication and coordination between project stakeholders, including testers, the development team, and company representatives, can lead to misunderstandings, misalignment, and delays in decision-making.
* Budget overruns: The budget allocated for the project can go over budget if costs are estimated incorrectly, there are unforeseen expenses, or insufficient resources are allocated.

**Product risks:**

* Incomplete or Inaccurate Content: Content on the Website may contain inaccuracies, incomplete details or outdated content, which may negatively impact the overall user experience.
* Compatibility Issues: The Website may have compatibility issues with some major web browsers, operating systems or devices. This may cause usability issues for some users.
* Security Vulnerability: Some app vulnerabilities can expose user data, compromising the security and trust of the platform.
* Performance issues: The website may experience delays in processing user actions, page loading delays or inadequate performance under high user traffic conditions.
* Functional Deficiencies: There is a possibility that the implemented functionalities may not meet the expected requirements, which may make the user unhappy or have a bad experience.

**2.2 Test analysis:**

* Assess business requirements and acceptance standards for relevant functionality.
* Defines the test method, test categories and techniques used.
* Describe the conditions that will be tested during the testing process.
* The testing method is functional testing of the UI (Graphic User Interface): is related to evaluating the functionality and behavior of an application. These tests examine whether the functions, buttons, input fields and other elements of the interface work correctly and give the user the results he expected.

**2.2.1 Techniques used for GUI (Graphical user interface) testing include:**

* **Positive testing**: check that the fields are complete correctly according to the requirements;
* **Negative testing**: check that fields cannot be filled with incorrect or invalid information;
* **Boundary value analysis testing**: check that the fields can accept the minimum and maximum values ​​allowed.
* **Equivalent Partitioning Testing:** this method involves dividing the input data into groups or partitions so that the important representatives in each group are tested.

**2.2.2 Test conditions for the "Programeaza-te" and “Contact” category**

1. **Evaluation of Displaying the Correct Interface:**

This condition focuses on ensuring a smooth user experience by displaying the interface properly.

1. **Verification of Required Field Marking:**

It is essential to ensure that all required parts of the form are properly marked to indicate that they are required.

1. **Recording of Programming with Correct Information:**

This condition checks if the schedule was successfully registered when the user enters all the data in the appropriate fields.

1. **Non-Recording of Programming with Incorrect Information:**

When the user enters incorrect or invalid data in the fields, we will check that the appointment is not recorded.

1. **Functionality of the “Termeni si conditii” link:**

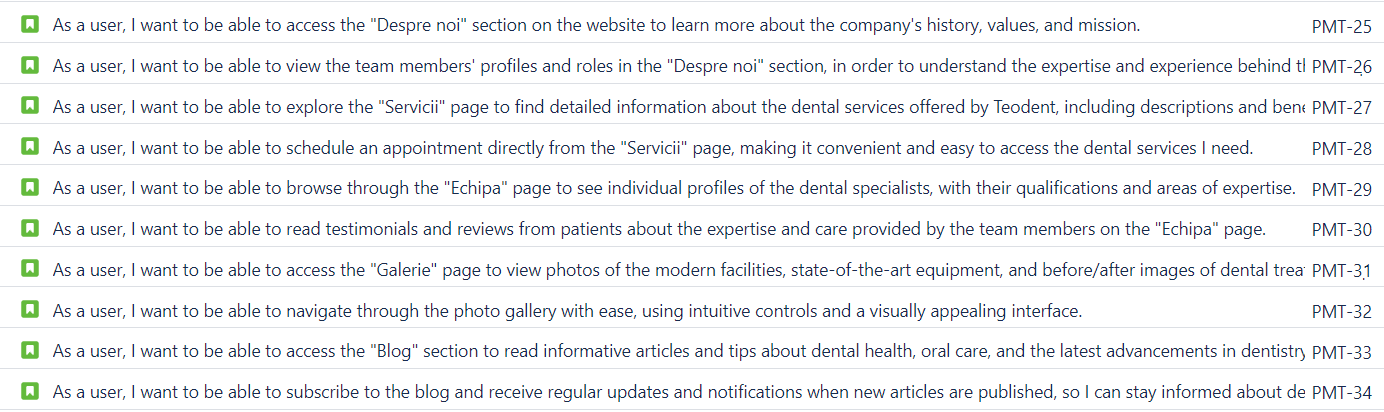
This condition requires verification that after clicking on the “Termeni si conditii” link, you have successfully opened a new page containing the terms and conditions. It should be easy for the user to return to the main page.

1. **Testing of the “Trimite” Button:**

This condition focuses on ensuring that the “Trimite” button works correctly and performs the intended functionality.

*#Note: Screenshots with test conditions list from Jira*

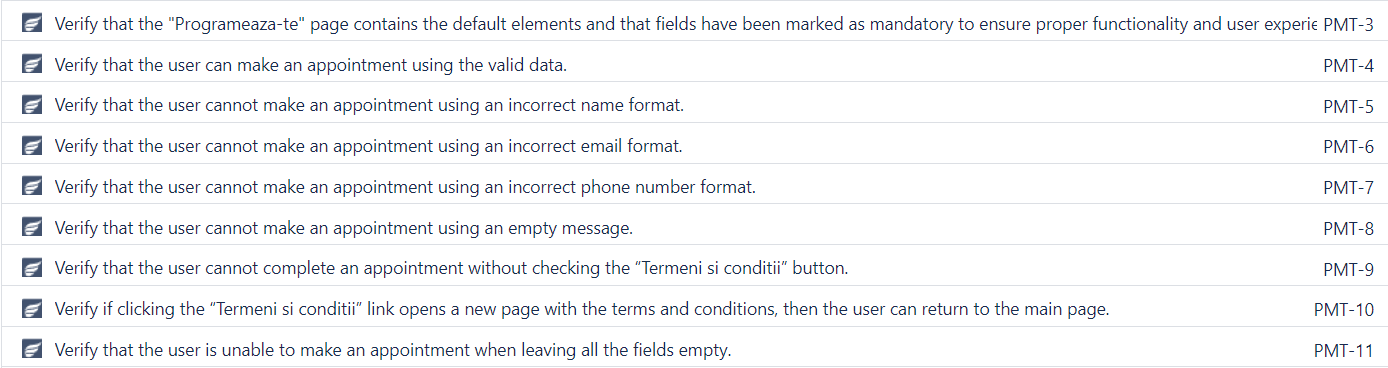
****

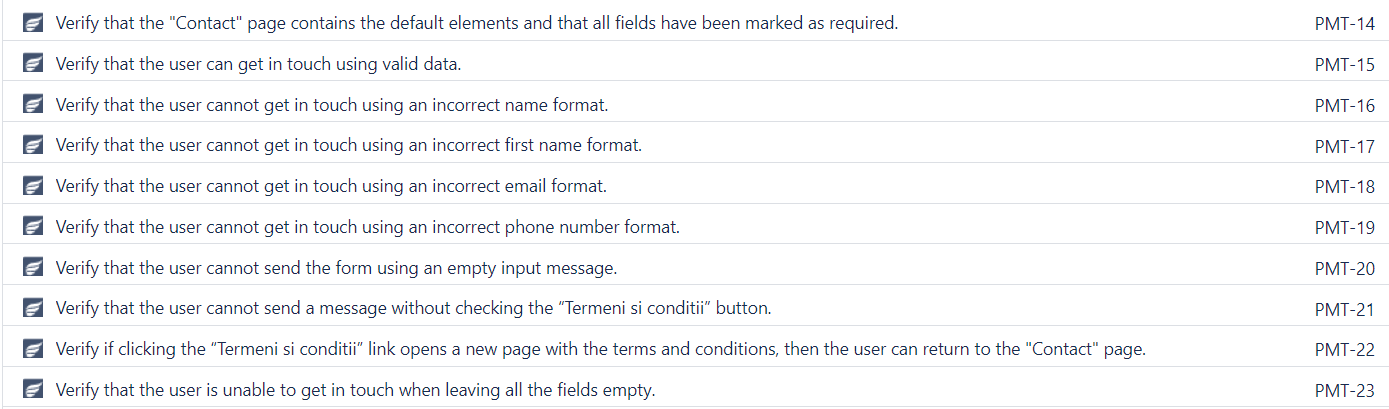
****

**2.3 Test design:**

* The purpose of the Zephyr Squad tool is to facilitate the creation of working test cases. Positive testing, negative testing, and use case testing are test design techniques used to create test cases based on specification analysis.
* Test scenarios and test cases were established according to the identified acceptance criteria.
* For each test case, the required data and expected results were established.
* Test cases were prioritized based on risk and importance.

*#Note: Screenshot with test cases from Jira*



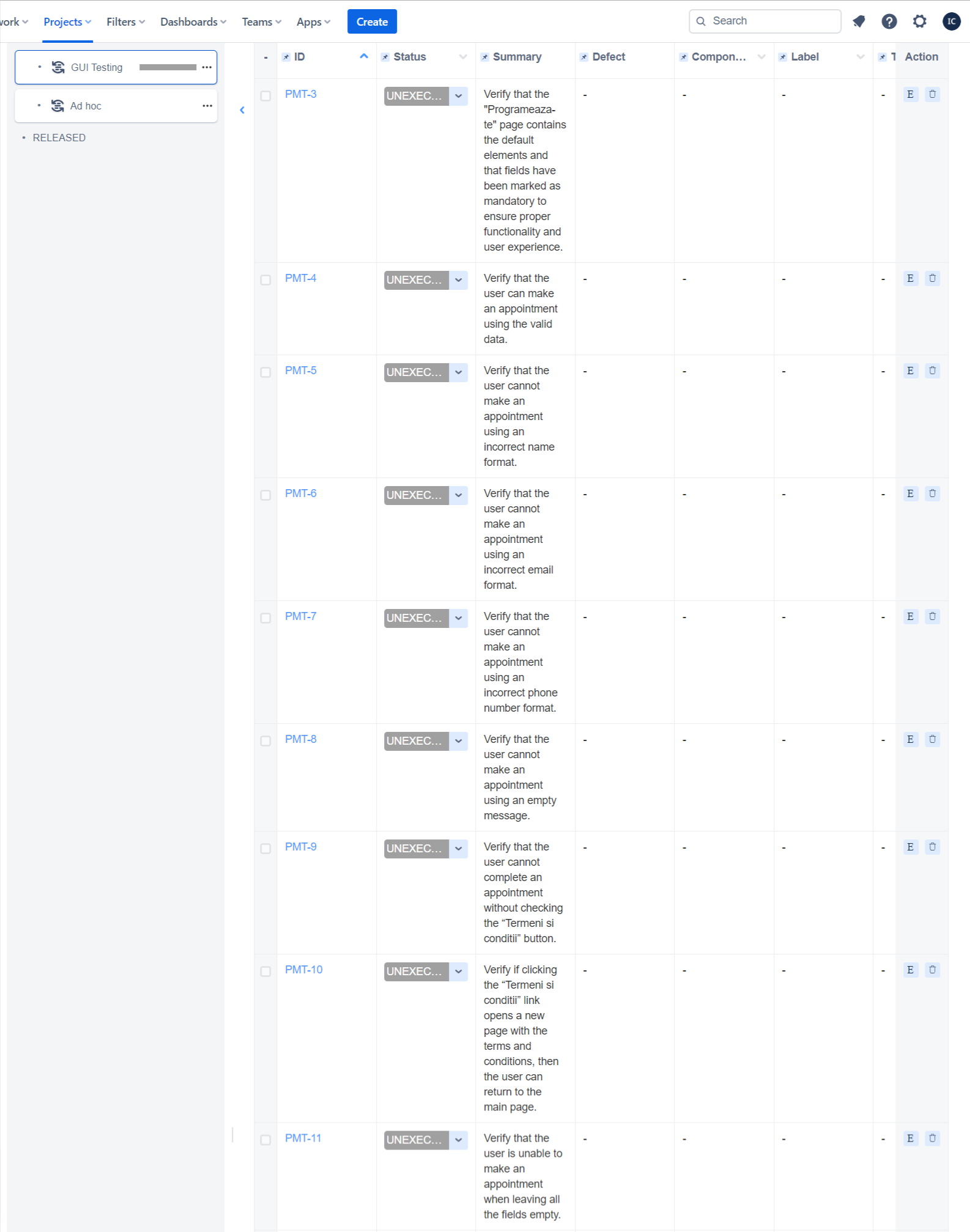


**2.4 Test implementation:**

Before starting the test execution phase, the following must be prepared:

* The test environment is up and running: <https://teodent.ro/>
* A cycle summary has been created called “GUI Testing”.
* The cycle summary now includes test cases.

*#Note: Screenshot with cycle summary from Zephyr*





**2.5 Test Execution:**

- The test cases are executed according to the test cycle summary that was created:*GUI Testing*

- We have prepared the test data that is required to run the test cases.

- Errors were identified based on the tests that were performed. The full bug report is available.

**2.6 Test Closure:**

- Execute the test cases according to the test plan and schedule.

- Observe actual and expected results.

- Record any problems or issues that arise during testing.

- To ensure that the defects were successfully resolved, we performed additional tests.

- Test closure reports should include test activities, results, and suggestions.

* 1. **Test monitoring and control:**

1. **Test deliverables:**

* **Test plan**
* **Test conditions**
* **Test cases**
* **Daily test summary reports**
* **Traceability matrix**
* **Test case results**
* **Bugs report**