“Calarim impreuna” version 1.0

Test Plan

January, 3, 2025

1. **Introduction**

This document summarizes all testing activities related to the product “Calarim impreuna” 1.0.

* 1. **Goals**

The goal of testing the "Calarim Impreuna" site is to ensure it delivers a seamless and localized user experience across various functionalities, including:

1. Validation of Core Functionalities: Verify that critical features like login and account management work as expected under different scenarios.
2. Localization Accuracy: Ensure that the application adapts seamlessly to multiple languages (Romanian, English, French), including error messages and interface elements.
3. Compatibility: Confirm that the application functions properly on supported browsers and operating systems.
4. Performance Under Load: Assess system responsiveness and behavior under heavy user or data loads.
5. Error Handling: Validate the consistency and appropriateness of error messages and responses for invalid inputs.
6. Regression Testing: Ensure that existing features remain functional after updates or changes.
7. Documentation Verification: Validate the accuracy and usability of online help and documentation.
8. Risk Mitigation: Identify and address potential issues like delays in localization or browser-specific behaviors.

By achieving these goals, the testing ensures that the "Calarim Impreuna" application is reliable, user-friendly, and meets the expectations of its diverse user base.

* 1. **Assumptions**

1. Application Readiness:
   * The "Calarim Impreuna" application is in a stable state for testing.
   * All dependent modules required for the testing scope are fully implemented.
2. Localization Preparedness:
   * All localization settings and translations for supported languages (Romanian, English, French) are complete and integrated into the application.
3. Test Data Availability:
   * Test data (valid and invalid user credentials) is available and prepared in alignment with the application's functional requirements.
   * Environment-specific configurations, such as user accounts, are in place.
4. Test Environment Setup:.
   * Supported browsers (Chrome, Firefox, Edge) are installed and configured for compatibility testing.
5. Documentation and Standards:
   * Detailed documentation of functional and localization requirements is accessible for cross-referencing during testing.
   * Standards for defect severity and acceptance criteria are predefined and agreed upon.
6. Team Roles and Communication:
   * All team members and roles, including test leads and automation programmers, are clearly defined and prepared to support the testing effort.
7. Testing Timelines:
   * The testing schedule aligns with the overall project plan, allowing sufficient time for test execution and bug fixing before deployment.

These assumptions ensure that testing progresses smoothly, focusing on the objectives outlined in the test plan. If any assumption is invalid, testing outcomes might be impacted.

* 1. **Risk and Assets**

Testing the "Calarim Impreuna" application involves risks and valuable assets. Key risks include delays in localization, which may affect multilingual support. Scheduling localization development early mitigates this risk. Browser compatibility issues could also arise, requiring cross-browser testing to ensure functionality across Chrome, Firefox, and Edge. Delays in resolving critical defects may slow progress, but assigning clear priorities and tracking resolutions can help. Limited test coverage for edge cases is another risk, which prioritizing high-risk areas can address. Performance bottlenecks under peak loads require early stress and volume testing to identify and optimize vulnerabilities.

* 1. **References**

The Test Plan has derived its content from the document “Specificatii”.

1. **Features to be tested**

The following features and functions of the product “Calaarim impreuna” 1.0, described in the document “Specificatii” should be tested:

* User Authentication (login functionality, localization support for login and error messages)
* Localization features (multi-language support, localization support for error messages)
* User Role Management ( role-based permissions for administrators, users and viewers; verification of restricted access based on user roles)
* Content Management ( ability to create, edit and delete user profiles; management of localized content and forms)
* Navigation and links ( functionality of key links like “Creeaaza un cont acum!” and “Programeaza-te acum”)
* Error handling ( proper display of error messages for incorrect inputs, localization of error messages across supported languages)
* Basic Reporting (logging user actions and system level avents, generating reports on user interactions and login attempts)

1. **Features not to be tested**

The following features and functions of the “Calarim impreuna” 1.0, described in the document “Specificatii” should not be tested:

* Database performance testing
* Detailed validation of visual design and styling, which are reviewed by the UI/UX team
* Mobile responsiveness
* Testing on outdated or unsupported browser version

1. **Approach**

The software testing often occurs in two phases during the development of an item: development testing and system testing phases. The System testing occurs after the item has been released by the developer from development testing. In this document there are using these two terms/definitions.

The required types of testing for the product, identified at the moment are:

* Volume Testing
* Cross Browser Testing;
* Security Testing;
* Installation Testing/Documentation;
* User access Testing;
* End to End integration testing;
* Specification adherence;
* Functional testing;

All these types of testing are prioritized, correspond to the Client team requirements.

* 1. **Cross Browser Testing**
     1. **Objectives**

The purpose of cross-browser testing is to verify that the “Calarim impreuna” 1.0 functions correctly across different browser types and versions.

* + 1. **Processing**

The cross-browser testing process for “Calarim impreuna” includes the following steps:

* Setup Testing Environment:
  + Ensure appropriate test servers are set up with required software (PHP, MySQL, Apache)
* Browsers to be tested:
  + Google Chrome ( latest stable version)
  + Mozilla Firefox ( latest stable version)
  + Microsoft Edge ( latest stable version)
* Compatibility focus:
  + Ensure all interactive features work seamlessly across browsers
  + Validate error message consistency in localized interfaced for each browser
    1. **Completion Test Criteria**

The cross-browser testing will be considered completed when the following criteria are met:

* All key features (login, localization, scheduling) work as expected on all tested browsers.
* No critical defects are found during normal flow execution on any tested browser.
  1. **Security Testing**
     1. **Objectives**

The purpose of security testing is to verify the robustness of “Calarim impreuna” 1.0 in protecting user data and system integrity. This includes assessing the correctness of user management features, validating protection against unauthorized access.

* + 1. **Processing**

In the security testing should be included the following testing tasks:

* Validation of user management
  + Verify that user roles and permissions are correctly implemented and enforced
  + Ensure that unauthorized users cannot access restricted sections of the site
* Isolation of user actions
  + Verify that user actions are isolated to their respective accounts and do not interfere with others
    1. **Completion Test Criteria**

The completion test criteria should be a complete execution (without any major issue) for all required testing tasks described above.

* 1. **User Access Testing**
     1. **Objectives**

The purpose of this task is to verify the correctness of the implementation of the client requirements regarding the user’s access in the product “Calarim impreuna” 1.0. This ensures that users are granted appropriate access based on their roles within the system.

* + 1. **Processing**

The process for the user access testing of the product “Calarim impreuna” 1.0 includes the following:

* Familiarize with the user access requirements (understand all documented user roles and access rights)
* Prepare the test scenarios in the product “Calarim impreuna” 1.0 :
  + Test individual user roles
  + Verify actions like “Programeaza-te!”, “Ai uitat parola?”, “Ajutor” page.

The user’s access testing will be a part of the functional testing of the product “Calarim impreuna” 1.0.

The testing environment should be included the product test server with required software.

* + 1. **Completion Test Criteria**

The completion test criteria should be a successful execution of the user testing access testing for the normal flow in the product “Calarim impreuna” 1.0.

* + 1. **Test Reports**

The test reports should be generated by the user access testing are:

* Summary Test Report: User’s Access Matrix
  + Provides a comprehensive view of all permitted actions across roles and access levels
  1. **End to End Test Criteria**
     1. **Objectives**

The purpose of this task is to verify the correctness of the implementation of the end-to-end workflow in the testing environment for the product “Calarim impreuna” 1.0.

* + 1. **Processing**

The process for end-to-end integration testing of the product “Calarim impreuna” 1.0 includes the following subprocesses:

Scenario 1: Booking Appointment (On-Demand Flow)

1. Create Users
   * Register new users with appropriate roles (e.g., Viewer, Author).
   * Verify data insertion in the database and validate the UI for the registered user.
2. Create Appointment
   * Log in as a registered user.
   * Navigate to the "Programează-te acum" page.
   * Select a horse, date, and time.
   * Submit the appointment request.
3. Validate Appointment
   * Check the database for appointment details.
   * Verify confirmation messages displayed in the user's selected language.
   * Validate that the appointment details are saved accurately.
4. View Scheduled Appointment
   * Log in and navigate to the "Programările tale" section.
   * Confirm that the scheduled appointment details are displayed correctly.
5. Modify Appointment
   * Update the appointment details (change date or service).
   * Verify updated details in the database and UI.

Scenario 2: Multilingual Workflow for Booking and Account Management

1. Create Users
   * Create accounts in multiple languages (e.g., Romanian, English, French).
2. Change Language
   * Navigate to the language dropdown and switch between languages.
   * Ensure that all interface elements update correctly.
3. Login and Book Appointment
   * Log in using localized credentials.
   * Verify that messages, labels, and buttons appear in the selected language.
4. Validate Localization
   * Test confirmation, error, and success messages in all supported languages.

Scenario 3: User Account Deletion

1. Validate Deletion
   * Ensure user data is removed from the database.
   * Confirm that deleted users can no longer log in.

Workflow scenario execution:

* Booking workflow: Test booking functionality end-to-end, including user registration, login, booking, editing and deleting appintments
* Localization Workflow: Test switching between supported languages and verifying that all features and messages are updated accordingly

The testing enviroment should be included in the product test server with the required software.

* + 1. **Completion Test Criteria**

The completion test criteria should be a successful execution of the end-to-end integration testing for the normal flow in the product “Calarim impreuna” 1.0.

* + 1. **Test Reports**

The test reports that should be generated by the end-to-end integration testing are:

* Summary test report: Booking Workflow
* Summary test report: User management
  1. **Localization Testing**
     1. **Objectives**

The purpose of this task is to verify the correctness of the implementation of localization functionality across the “Calarim impreuna” platform. This ensures that all user interface elements, messages and workflows can adapt to the selected language (Romanian, English, French) and that the functionality remains intact in localized environments.

* + 1. **Proccessing**

The process for localization testing of the product “Calarim impreuna” 1.0 includes the following subprocesses:

Scenario 1: Language Switching

1. Access the Language Dropdown
   * Verify the visibility and usability of the language menu on all pages.
2. Select a Language
   * Switch between Romanian, English, and French.
   * Validate that the page reloads and all UI elements reflect the selected language.
3. Validate Critical Pages
   * HomePage.html
   * Programari.php
   * LogIn.php
   * contact.html
   * educational.html
   * galerie.html
4. Check Non-Translated Elements
   * Verify that all texts, labels, buttons, and messages are translated correctly and no untranslated strings appear.

Scenario 2: Multilingual Account Management

1. Localized User Creation
   * Register new accounts using each language.
   * Verify that registration confirmation messages and errors are displayed in the selected language.
2. Localized Login
   * Log in using localized credentials.
   * Validate that the UI reflects the selected language after login.
3. Localized Error Messages
   * Input incorrect credentials and validate that error messages are displayed in the correct language.

Scenario 3: Appointment Workflow Localization

1. Navigate to "Programează-te acum"
   * Verify the page elements (labels, buttons, instructions) in all supported languages.
2. Complete Booking
   * Select a horse, date, and time while ensuring all messages and labels reflect the selected language.
3. Error Handling
   * Test validation errors (missing required fields) and ensure that error messages are displayed in the correct language.

Scenario 4: Static and Dynamic Content Localization

1. Static Content
   * Verify that all static text on pages such as "Știai că?" (educational.html) and "Galerie" (galerie.html) is translated correctly.
2. Dynamic Content
   * Validate that dynamically generated text, such as confirmation messages, appointment details, and user-specific information, appears in the selected language.

* Workflow Scenario Execution:
  + Language switching: validating transitions between languages
  + Localized Workflow: ensuring that end-to-end workdlows are fully functional in any language
  + Error messages: confirming appropriate localization of errors in all workflows

The testing enviroment should be included the product test server with required software and the client workstation with the required software.

* + 1. **Completion Test Criteria**

The completion test criteria should be a successful execution of the localization testing for the normal flow in the product “Calarim impreuna” 1.0.

* + 1. **Test Reports**

The reports that should be generated by the localization testing are:

* Summary test report: Language Switching (switching language across all pages)
* Summary test report: Localized Workflows (results of testing user registration, login and booking workflows in all supported languages)
* Summary test report: Error Handling ( documentation of localization in error messages and validation workflows)
* Summary test report: Consistency and Fallback ( evaluation of terminology consistency and fallback behavior in case of missing translations)
  1. **Specification adherence**

The main purpose of this task is to ensure that the product “Calarim impreuna” 1.0 meets all client requirement as detailed in the specification document. Every implemented functionality should align precisely with the stated expectations.

* 1. **Functional testing**

For the product “Calarim impreuna” 1.0 there are planned the following types of testing:

* Development Testing: Execution of the test case suite for validating individual functionalities
* System Testing: Execution of end-to-end workflows and scenarios
  + 1. **Objectives**

The main objective of the functional testing is to validate the implementation of “Calarim impreuna” 1.0 platform’s functionality against the client requirements outlined in the specifications document. Additionally, the testing will ensure compliance with industry standards and verify that normal workflows and edge cases (e.g. incorrect password during login) are correctly handled.

* + 1. **Scope**

All major components of the “Calarim impreuna” 1.0 will be tested. The scope includes:

* User Registration and Authentication
* Localization
* Appointment Booking
* User Management

The set of the designed test cases will test the whole normal flow functionality (with valid input data) and some typical exceptions (incorrect password during login). Each test case is uniquely identified and describes specific actions performed on platform objects (user, appointment). Test cases for related components (appointment booking) are grouped intro comprehensive scenarios.

* + 1. **Processing**

The functional testing process includes the following subprocesses:

Scenario 1: User Registration Workflow

1. Access the Registration Page
   * Navigate to sign\_up.php and verify the presence of the registration form.
2. Submit Valid Details
   * Enter all required fields (e.g., username, email, password) with valid data and submit.
3. Verify Registration Success
   * Confirm the success message is displayed in the UI.
   * Check the database to ensure the user details are correctly stored.
4. Submit Invalid Details
   * Test missing or invalid fields (e.g., invalid email format, mismatched passwords).
   * Ensure appropriate error messages are displayed.

Scenario 2: User Login Workflow

1. Access the Login Page
   * Navigate to LogIn.php and verify the presence of the login form.
2. Log in with Valid Credentials
   * Enter correct username and password.
   * Verify successful login and redirection to the homepage.
3. Log in with Invalid Credentials
   * Enter incorrect credentials.
   * Confirm that the error message ("Username or password incorrect") appears.

Scenario 3: Booking Appointment Workflow

1. Navigate to "Programează-te acum" Page
   * Ensure the page loads correctly with all necessary form fields.
2. Submit Booking with Valid Details
   * Fill out the form with valid data (service type, date, and time) and submit.
3. Validate Booking Confirmation
   * Verify the confirmation message appears on the UI.
   * Check the database to ensure the appointment is recorded accurately.
4. Submit Booking with Invalid Details
   * Leave required fields empty or enter invalid values.
   * Confirm that appropriate validation errors are displayed.

Scenario 4: Language-Specific Functional Workflows

1. Localized Registration
   * Complete the registration process while using each supported language (Romanian, English, French).
   * Verify that all form labels, messages, and buttons appear in the selected language.
2. Localized Login
   * Log in using localized credentials and validate the UI elements and messages.
3. Localized Booking
   * Complete a booking while the language is set to French or English.
   * Ensure that confirmation and error messages are correctly translated.

Scenario 5: Account Deletion and Management

1. Admin Deletes User
   * Log in as Manager and delete a user account.
   * Confirm that the user can no longer log in and their data is removed from the database.
2. Validate Asset Reassignment
   * If the deleted user had appointments, verify that the assets are reassigned or flagged for review.

Workflow Scenario Execution:

* User Registration
* User Login
* Booking Workflow
* Language-specific Testing

The testing environment should be included product test server with the required software.

* + 1. **Completion Test Criteria**

The functional testing will be considered complete when:

* A checklist covering all test scenarios and test cases has been executed
* All test cases have been completed successfully or have been appropriately addressed
* The successful completion of functional testing serves as the trigger for the deployment process.
  + 1. **Test Reports**

The reports for the functional testing that should be generated by the product testing are:

* Bugs/Issues summary report – a full list of all identified bugs/issues during the developer/system testing, their severity/priority, their current status ( identified, in progress, fixed, verified)
* Summary test reports – total numbers of test cases/scenarios, percentage of test cases executed and passed, comments and notes for failed cases

1. **Testing. Resources. Schedule.**

This section describes significant information of the testing effort. It items such as roles that are involved in testing on the project, the responsibilities of each, staffing information, etc.

* 1. **Staffing**

| **Name** | **Role** | **Involvment** | **Responsability** |
| --- | --- | --- | --- |
| Porumb Denisa | QA lead | Full time | Prepare test plan, provide direction for and review test cases; design and execute test cases; report identified bugs/issues; verify fixed bugs; control tests execution and reporting |
| Siciu Alina | QA engineer | Part-time | Design and execute test cases; report identified bugs/issues; verify fixed bugs |
| Patko Stefania | QM | Full-time | Review all test documents ( test plan, test cases, test reports) and controls the daily activities of the QA team |
| Nenisca Antonia | QA engineer | Part-time | Design and execute test cases; report identified bugs/issues; verify fixed bugs |

* 1. **Schedule**

The test milestones of the project depend on the project timeline – these milestones are synchronized with the main development tasks.



