PET PALS

pet adoption & pet sitting

PetPal IEEE 829 Test Plan

Version 1.0

26/10/2024

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Najah Ismail	25/10/2024	Mishra Apurva	26/10/2024	Test Plan

Test Plan Identifier

Master Test Plan 1.0

Introduction

The document specifies the master test plan for PetPal web application. The plan covers the 4 main features: user profile, adoption, pet sitting, and events. The aim of the tests is to ensure correct functioning and consistent results.

1. Test Items

The testing coverage is separated in Front-End and Back-End testing with specific members tagged to each testing section for maximise team productivity.

Front-End	User Interface - Ensure all UI elements are present as per the system requirements for each Use-Case Form Validation - Ensure that all inputs are present Navigation: - Ensure Navigation flows for each feature are present and work smoothly. Authentication - Ensure that Authentication is functioning for secure access management.
Front-end and back-end integration	1. Ensure inputs filled by user are correctly posted through the database through POST requests 2. Ensure complete and correct data retrieved with GET requests and shown on the UI
Back- End	Database Interactions: - Ensure that the data is properly persisted into the database API Error Handling - Validate API for successful operations - Validate error handling for invalid API calls.

2. Features To Be Tested

2.1 Front-end Testing

Use Jest as the test runner and testing library for writing unit and integration tests as it is a powerful combination for testing JavaScript applications.

2.2 Back-end Testing

Unit testing was conducted to validate server responses, API endpoints, and database interactions to ensure they behave as expected under different conditions.

Using Pytest, we tested individual API endpoints to ensure they handle requests correctly, return expected status codes, and manage both valid and invalid inputs effectively. The straightforward syntax and central testing mechanisms helped streamline testing and debugging, ultimately improving code quality and reliability.

Function	Test Cases
Users	• Get Empty User List • Get User List • Get User by email • Get User by invalid email • Create User • Create User with existing email • Update User • Update User by email
Pets	• Get pets lists • Create pet
Events	• Get empty events list • Get all events • Get event by id • Get event by invalid id • Create event • Create event with missing data • Update event • Update event with invalid id • Delete event • Delete event with invalid id • Get event by User • Get events by User Interests
Event Interest	• Get all events interest • Get event interests by event id

	• Get event interest by invalid id • Create event interest • Create event interest with missing data • Update event interest • Update event interest with invalid id • Delete event interest • Delete event interest with invalid id
Adoption	• Get empty adoption list • Get all adoptions • Get adoption by ID • Get adoption by invalid ID • Create adoption • Create adoption with missing data • Update adoption • Update adoption with invalid ID • Delete adoption • Delete adoption with invalid ID • Get adoption by User • Get adoptions by User Interests
Adoption Interest	• Get empty adoption interest list • Get all adoption interests • Get adoption interest by ID • Get adoption interest by invalid ID • Create adoption interest • Create adoption interest with missing data • Update adoption interest • Update adoption interest with invalid ID • Delete adoption interest • Delete adoption interest with invalid ID
Sitter Interest	• Get empty sitter interest list • Get all sitter interests • Get sitter interest by ID • Get sitter interest by invalid ID • Create sitter interest • Create sitter interest with missing data • Update sitter interest • Update sitter interest with invalid ID • Delete sitter interest

	• Delete sitter interest with invalid ID
Sitting Request	• Get empty sitting request list • Get all sitting requests • Get sitting request by ID • Get sitting request by invalid ID • Create sitting request • Create sitting request with missing data • Update sitting request • Update sitting request with invalid ID • Delete sitting request • Delete sitting request with invalid ID • Get sitting request by User • Get sitting requests by User Interests

3. Features Not To Be Tested

The following cases are not covered under our test plan:

- Third-party libraries and dependencies unless they directly impact core application functionality or if a conflicting release/disruption is noted.
- Non-functional requirements such as performance and load testing.

4. Approach

The following test levels are conducted:

4.1 Unit Testing

Individual components are tested to ensure that they function as expected and handle inputs and errors. An in memory database and mock data is used to conduct the above.

4.2 Integration Testing

Validate workflows that span multiple modules and between the frontend and backend server.

4.3 System Testing

End-to-end testing of the application is tested by drawing on each use-case i.e. User authentication, Events, Sitters and Adoption.

5. Item Pass/Fail Criteria

5.1 Pass

A test case is declared passed if the actual results are aligned with the expected results.

5.2 Fail

A test case is declared failed if there are unexpected results, errors or incorrect data handling. If the failure of the test case has implications in another Test Case it is also categorized as Critical Fail.

5.3 Critical Fail

Any test-case categorized as Fail that may cause system crashes or illegal data operations.

6. Suspension Criteria And Resumption Requirements

Suspension Criteria	Testing is suspended if any Test Cases are Categorized as Critical Fail (5.3). The suspension will occur in either Front-end, backend or full team depending on the Test Case.
Resumption Requirements	Testing is resumed once the patch is released to fix the error and the Test Case is categorized as Pass (5.1).

7. Test Deliverables

7.1 Test Plan

The IEEE 829 Test Plan document which details the testing protocol.

7.2 Test Cases

Detailed test cases for System testing (4.2) each feature and functionality.

7.3 Unit Test Code

The Unit Test for each functionality will be updated and made available to the team members through the Github repository.

8. Testing Tasks

8.1 **Test Case Development**:

Write test cases for each identified feature and use-case requirement.

8.2 **Setup Test Framework**:

The testing frameworks for the Front-end and Back-end are configured and set up to maintain consistency in the testing protocol.

8.3 Test Log

The output of each test case is logged to ensure no Test Cases are categorized as Fail or Critical Failure without appropriate countermeasures.

9. Environmental Needs

9.1 Database

The back-end server must be configured and set-up for each member.

9.2 Testing Tools

The testing framework as detailed in (2.) must be accessible to each member of the testing team.

10. Responsibilities

Role	Responsibility
QA Manager	Oversee test planning, execution and reporting.
QA Engineer	Develop and define test cases.
Lead Developer	Oversee and review test case results.
Project Manager	Approve final test plan and review test summary reports.

11. Staffing And Training Needs

The testing team has taken on the following 2 roles to conduct thorough and efficient testing:

11.1 Test Engineers: Familiarity with Pytest, Jest, Postman, and web application testing.

11.2 Developers: For defect resolution and understanding test results.

12. Schedule

Task	Start Date	End Date
Test Planning	18/10/2024	20/10/2024
Test Case Development	20/10/2024	22/10/2024
Test Environment Setup	22/10/2024	23/10/2024
Test Execution	23/10/2024	26/10/2024
Defect Resolution		
Test Summary Reporting	26/10/2024	26/10/2024
Final Review and Approval	26/10/2024	26/10/2024

13. Risks And Contingencies

13.1 Inconsistency in testing frameworks and approach.

The testing framework and methodology is agreed upon by the team members before the commencement of Test Execution.

13.2 Delayed Resolution

Any errors discovered during testing are resolved by the team member in charge. Incase of capacity issues, we follow a Transfer protocol and the responsibility is delegated.

Approvals

Approved by Project Manager : Mishra Apurva

Date : 26/10/24

Signature : Apurva

Role: Product Manager