

## **Assignment #1 - Test Plan**

Names: Neha Annareddy, Vaishakh Nair, Minha Raza, Reese Rapin, Ayush Sunkad

### **Part I. Description of Overall Test Plan**

Our test plan focuses on making sure that our app is functional and reliable. We also plan to test the performance, functionality and usability of the platform. We plan on using multiple testing strategies such as unit testing, integration testing, functional testing and performance testing. 3 of our group members have taken a Quality Assurance class in our past semesters and we plan to use that knowledge to make sure our app is well tested and robust. We plan to use Blackbox and Whitebox testing for our features and code. Our normal cases will validate expected behavior, and abnormal cases will assess system robustness.

Our testing process includes Unit testing for verifying individual routes in Flask and our helper functions. Our integration testing helps to make sure there is seamless communication between our frontend, backend, APIs and database. Our Functional testing helps us to validate the user creation and login process. We also plan to further include performance testing and security testing as we get closer to the expo.

### **Part II. Test Case Descriptions**

TC01 - Like a Post Test

TC01.1 Purpose of Test: To verify that users can successfully like a post and that the like count updates correctly.

TC01.2 Description of Test: A logged-in user attempts to like a post on the platform. The system should record the like and update the like count accordingly.

TC01.3 Inputs Used for Test: Logged-in user and valid post ID

TC01.4 Expected Outputs/Results:

The post's like count increases by 1.

The user's like is recorded in the database.

The like button updates visually to indicate the post is liked.

If the user unlikes the post, the like count decreases by 1.

TC01.5 Normal/Abnormal/Boundary Case Indication:

Normal: A user likes a post successfully.

Abnormal: A user attempts to like a non-existent post.

Boundary: Rapidly liking and unliking a post multiple times.

TC01.6 Blackbox/Whitebox Test Indication: Blackbox

TC01.7 Functional/Performance Test Indication: Functional

TC01.8 Unit/Integration Test Indication: Unit

---

TC02 Data Privacy and security Test

TC02.1 This test will ensure that sensitive user data is securely transmitted.

TC02.2 This test will check if sensitive data is encrypted, and unauthorized access prevention measures exist.

TC02.3 User and breeder credentials.

TC02.4 Data is encrypted and unauthorized access of any kind is blocked.

TC02.5 Normal

TC02.6 Blackbox

TC02.7 Security

TC02.8 Integration

---

TC03 Performance and Load Test

TC03.1 To assess the performance of the app during high load and peak usage times.

TC03.2 To simulate high number of users using the website

TC03.3 Simulating user interactions of high frequency and volume.

TC03.4 The website remains stable and functional.

TC03.5 Normal

TC03.6 Blackbox

TC03.7 Performance

TC03.8 Integration

---

TC04 Cross-Platform compatibility and screen size responsiveness test for web

TC04.1 To ensure that the website provides a consistent, functional user experience across various browsers, devices and screen sizes

TC04.2 We will test the functionality, layout and general usability across different browsers and screen resolutions.

TC04.3 Combination of different web browsers (Chrome, Safari, Edge), operating systems (Windows, MacOS, Linux), and screen resolutions (360\*640, 768\*1024, 1280\*720, 1920\*1080)

TC04.4 The website should adjust its layout appropriately to fit different screen sizes without any usability issues and readability.

TC04.5 Normal

TC04.6 Blackbox

TC04.7 Functional

TC04.8 Integration

---

TC05: User Testing the Login Functionality

TC05.1 Verify that all the users can successfully log in with the valid credentials.

TC05.2 The system should authenticate users with a valid username and password.

TC05.3 Possible inputs can be Username: valid\_username, Password: valid\_password

TC05.4 User is successfully logged in and redirected to the website's appropriate dashboard.

TC05.5 Normal

TC05.6 Blackbox

TC05.7 Functional

TC05.8 Integration

---

TC06: User Testing the Invalid Login Attempt

TC06.1 Ensure that the incorrect credentials result in a failed login attempt and possibly keep track of the number of failed attempts for security purposes.

TC06.2 The system should reject all the invalid credentials and display an error message.

TC06.3 Possible inputs can be Username: invalid\_username, Password: wrong\_password

TC06.4 The user remains on the login page with an error message displaying. Also possibly adding a # of attempts left depending on the number of failed attempts.

TC06.5 Abnormal

TC06.6 Blackbox

TC06.7 Functional

TC06.8 Integration

---

TC07: Handling the downtime of the Website

TC07.1 Ensure that the system will handle the unexpected website downtime gracefully. By possibly having a message stating website will be back in some time.

TC07.2 If the website becomes inaccessible due to server failure, maintenance, or any other issues, then the user should receive an appropriate error message instead of just a black screen or generic browser error.

TC07.3 Possible inputs are stimulating the website downtime by shutting down the server or disconnecting from the internet to see what it does.

TC07.4 The user sees a custom error page stating, "Our website is temporarily down. Please check back later."

TC07.5 Abnormal

TC07.6 Blackbox

TC07.7 Functional

TC07.8 Integration

-----

TC08: Handling a Slow response time

TC08.1 Ensure that the website remains responsive and provides feedback to users during the slow server response times so that the users are not just waiting without any cause.

TC08.2 When the server takes longer than usual to load the data, the application should display a loading indicator and prevent timeouts from negatively affecting the user experience.

TC08.3 Possible inputs can stimulate slow response by artificial latency (ex. Delay API responses by 10+ seconds).

TC08.4 A loading spinner or message like "Fetching nearby cats, please wait..." which appears until the data is retrieved. If the response takes longer than 30ish seconds, then display an error message: "We are experiencing delays. Please try again later."

TC08.5 Boundary

TC08.6 Blackbox

TC08.7 Performance

## TC08.8 Integration

---

### TC09 - Add a New Pet Sub-Account Test

TC09.1 Purpose of Test: Ensure that users can successfully add a new pet sub-account under their main profile.

TC09.2 Description of Test: A logged-in user attempts to create a new pet sub-account by providing the required pet details. The system should associate the pet with the user's main account and display it in their profile.

TC09.3 Inputs Used for Test: Logged-in user, Pet name, Pet type (e.g., dog, cat), Pet breed (optional), Date of birth (optional), Profile picture (optional)

TC09.4 Expected Outputs/Results: The pet sub-account is successfully created and linked to the user's main profile, The pet appears in the user's profile under "My Pets.", The pet's details are stored in the database, If a profile picture is uploaded, it is displayed correctly.

TC09.5 Normal/Abnormal/Boundary Case Indication:

Normal: A user adds a pet with valid details.

Abnormal: A user tries to add a pet without a name.

Boundary: Adding the maximum allowed number of pets.

TC09.6 Blackbox/Whitebox Test Indication: Blackbox

TC09.7 Functional/Performance Test Indication: Functional

TC09.8 Unit/Integration Test Indication: Integration

---

### TC010 - Add a Friend Test

TC010.1 Purpose of Test: Ensure that a user can successfully send and accept a friend request on the platform.

TC010.2 Description of Test: A logged-in user attempts to send a friend request to another user. The recipient should receive a notification and be able to accept or decline the request. Once accepted, both users should appear in each other's friend lists.

TC010.3 Inputs Used for Test: Logged-in user (User A), Target user (User B), Friend request action (send, accept, decline),

TC010.4 Expected Outputs/Results: A friend request notification is sent to User B.

User B can accept or decline the request.

If accepted, both users appear in each other's friend lists.

If declined, the request is removed from the system.

TC010.5 Normal/Abnormal/Boundary Case Indication:

Normal: User A sends a request, and User B accepts it.

Abnormal: User A tries to send a request to a non-existent or blocked user.

Boundary: User A repeatedly sends friend requests to the same user.

TC010.6 Blackbox/Whitebox Test Indication: Blackbox

TC010.7 Functional/Performance Test Indication: Functional

TC010.8 Unit/Integration Test Indication: Integration

**Part III. Test Case Matrix: summarizes the test case coverage (items 1, 6-9 in a tabular format)**

Test Case ID	Abnormal/Normal/Boundary	Blackbox/Whitebox	Functional/Performance	Unit/Integration
TC01	Normal, Abnormal, Boundary	Blackbox	Functional	Unit
TC02	Normal	Blackbox	Security	Integration

TC03	Normal	Blackbox	Performance	Integration
TC04	Normal	Blackbox	Functional	Integration
TC05	Normal	Blackbox	Functional	Integration
TC06	Abnormal	Blackbox	Functional	Integration
TC07	Abnormal	Blackbox	Functional	Integration
TC08	Boundary	Blackbox	Performance	Integration
TC09	Normal, Abnormal, Boundary	Blackbox	Functional	Integration
TC10	Normal, Abnormal, Boundary	Blackbox	Functional	Integration