Test Plan for Veterinary Clinic

Asmaa Maher, Maya Walid, Nada Mohamed, Nermein Mohamed, Sara Yasser
Supervised by: Dr.Salwa Osama, Eng. Nada Ayman, Eng. Bassant Ahmed

May 28, 2024

Table 1: Document version history

Version	Date	Reason for Change
1.0	15-March-2024	Test Plan First version is defined.

GitHub: https://github.com/Veterinary-Clinic/vet_clinic

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describes the testing approach and overall framework that will drive the testing of our project. this doucment introduces:

- Defining Testing Objectives: It outlines the objectives and goals of the testing process, such as ensuring the functionality of systems like appointment, patient and doctor management, etc.
- Scope: It defines the scope of testing, including which features, functionalities, and modules will be tested. This ensures comprehensive coverage of the clinic's software systems.
- Testing Approach: It outlines the approach to be followed during testing, It also specifies the testing techniques to be employed.

2 Test Scenario 1

- **Users' Actions and Behaviors:**
  The user Login with his personal information to the website
- **Technical aspects of the requirement**
  - Database Management: The system should utilize a strong database management system to store and manage user's data efficiently. Implement mechanisms such as constraints, validations, and referential integrity to prevent data corruption, duplication, or inconsistency.
  - Electronic Medical Records:Secure storage and recovery of patient records. Access control mechanisms to ensure privacy.
  - Appointment System: Ability of handling many appointments.
- **Possible Scenarios of System Failure**
  - Data Loss: The loss of patient information can lead to disturbances in the operation of the vet clinic
  - Lack of Security: Unauthorized access to patient profile or their data can lead to privacy violations and stealing identity

2.1 Test Cases

Test Cases for the scenario mention in section 2 shown in Table 2

Table 2: Test Cases for Scenario 1

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC01	Valid credentials - Successful login	U01	Enter valid username and password then Click on "Login" button	Logged in succesfully
TC02	Invalid username - Failed login	U01	Enter invalid username and password then Click on "Login" button	failed Login
TC03	Invalid password - Failed login	U01	Enter valid username and invalid password then Click on "Login" button	failed Login
TC04	Empty username field - Failed login	U01	Enter valid username and leave password field empty then Click on "Login" button	user must should a message says password is empty
TC05	empty password field - Failed login	U01	leave username field empty and enter valid password then Click on "Login" button	user must should a message says username is empty

3 Test Scenario 2

- **Users' Actions and Behaviors:**
  The user books appointment with the doctor that he wants.

- **Technical aspects of the requirement**

  - Database Management: The system should utilize a strong database management system to store and manage user's data efficiently. Implement mechanisms such as constraints, validations, and referential integrity to prevent data corruption, duplication, or inconsistency.

- **Possible Scenarios of System Failure**

  - Data Loss: The loss of patient's appointment can lead to disturbances in the operation of the vet clinic
  - Lack of Security: Unauthorized access to patient profile or their data can lead to privacy violations and stealing identity
  - Bugs: This error can cause unexpected actions, such as incorrect appointment scheduling.

3.1 Test Cases

Test Cases for the scenario mention in section 9 shown in Table 9

4 Test Scenario 3

- **Users' Actions and Behaviors:**
  The user add his pet informations

- **Technical aspects of the requirement**

  - Database Management: The system should utilize a strong database management system to store and pet's data efficiently. Implement mechanisms such as constraints, validations, and referential integrity to prevent data corruption, duplication, or inconsistency.

- **Possible Scenarios of System Failure**

  - Data Loss: The loss of patient pet's information can lead to disturbances in the operation of the vet clinic
  - Lack of Security: Unauthorized access to patient profile or their data can lead to privacy violations and stealing identity
  - Software error and bugs: This error can cause unexpected actions, such as not adding the pet or data inconsistencies.

Table 3: Test Cases for Scenario 2

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC06	Successful appointment booking	U03	Login to the system then Navigate to the appointment booking page then Provide required details and Click on "Book Appointment" button	successfully booked
TC07	Failed appointment booking	U03	Login to the system then Navigate to the appointment booking page then leave one or more required details and Click on "Book Appointment" button	user should recieve error message says this field need to be filled
TC08	Failed appointment booking	U03	Login to the system then Navigate to the appointment booking page then select slot that is already booked and Click on "Book Appointment" button	user should recieve error message says this slot is already booked

4.1 Test Cases

Test Cases for the scenario mention in section 9 shown in Table 9

5 Test Scenario 4

- **Users' Actions and Behaviors:**
  The doctor updates his profile
- **Technical aspects of the requirement**
  - Database Management: The system should utilize a strong database management system to store and pet's data efficiently. Implement mechanisms such as constraints, validations, and referential integrity to prevent data corruption, duplication, or inconsistency.
  - User Authentication: Ensure that only authenticated users are allowed to access and modify their profiles.
- **Possible Scenarios of System Failure**
  - Data Loss: The loss of patient profile's information can lead to disturbances in the operation of the clinic
  - Input Data Validation Failure:The system does not properly validate user input data, leading to errors in the profile update process.

Table 4: Test Cases for Scenario 3

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC09	Successfully pet addition	U11	Login to the system then Navigate to the profile page then add pet page then add the required details and Click on "Add pet" button	successfully added
TC10	Failed pet addition	U11	Login to the system then Navigate to the profile page then add pet page then leave one or more required details and Click on "Add pet" button	user must should error message says this field need to be filled

- Software error and bugs: This error can cause unexpected actions, such as not editing the profile or data inconsistencies.

5.1 Test Cases

Test Cases for the scenario mention in section 9 shown in Table 9

Table 5: Test Cases for Scenario 4

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC011	Valid profile update	D10	Login to the system then to the profile page and navigate to the edit button and modify the fields and Click on "save" button	successfully updated
TC012	Cancel profile update	D10	Login to the system then to the profile page and navigate to the edit button and modify the fields and Click on "cancel" button	the profile remains the same without changes
TC013	Profile update with invalid input	D10	Login to the system then to the profile page and navigate to the edit button and modify the fields with invalid input and Click on "Save" button	Error message displayed

6 Test Scenario 5

- **Doctors' Actions and Behaviors:**
  The doctor add appointment

- **Technical aspects of the requirement**

  - User Interface (UI): Design an intuitive and user-friendly interface for doctors to add appointment schedules. Consider factors such as ease of use, accessibility, and responsiveness across different devices.
  - Database Management: Design a strong data model to represent appointment schedules, including fields such as date, time, and any relevant notes. The system should utilize a strong database management system to store and pet's data efficiently. Implement mechanisms such as constraints, validations, and referential integrity to prevent data corruption, duplication, or inconsistency.

- **Possible Scenarios of System Failure**

  - Data Loss: The loss of doctor's appointment can lead to disturbances in the operation of the clinic
  - Session Timeout: The doctor's session expires due to inactivity while modifying a schedule, causing them to lose unsaved changes and requiring them to log in again
  - Software error and bugs: This error can cause unexpected actions, such as not editing the profile or data inconsistencies.

6.1 Test Cases

Test Cases for the scenario mention in section 9 shown in Table 9

Test Cases for the scenario mention in section 9 shown in Table 9

Table 6: Test Cases for Scenario 5

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC014	Successful addition of a new appointment	D01	Login to the system then to the profile page and navigate to the add appointment section and fill the fields then Click on "Add appointment" button	successfully added
TC015	Adding appointment with missing or invalid information	D01	Login to the system then to the profile page and navigate to the add appointment section and leave one or more field empty then Click on "Add appointment" button	error message of validations displayed with invalid input
TC016	Adding appointment with wrong hours	D01	Login to the system then to the profile page and navigate to the add appointment section and add an end time slot after the start time then Click on "Add appointment" button	Error message displayed

7 Test Scenario 6

- **Admin' Actions and Behaviors:**
  The admin add doctor

- **Technical aspects of the requirement**

  - User Interface (UI): Design an intuitive and user-friendly interface for admins to add doctors. Consider factors such as ease of use, accessibility, and responsiveness across different devices.
  - Database Management: Design a strong data model to store doctor's information. The system should utilize a strong database management system to store data efficiently. Implement mechanisms such as constraints, validations, and referential integrity to prevent data corruption, duplication, or inconsistency.

- **Possible Scenarios of System Failure**

  - Data Loss: The loss of doctor records due to database corruption, accidental deletions, or software failures.
  - Session Timeout: the admin's session expires due to inactivity while they are in the process of adding a new doctor.
  - Software error and bugs: software bugs cause unexpected behaviors such as failure to save the doctor information, incorrect data being stored, or data inconsistencies.

7.1 Test Cases

Test Cases for the scenario mention in section 9 shown in Table 9

Table 7: Test Cases for Scenario 6

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC017	Successful addition of a new doctor	A02	Login to the system then to the dashboard and navigate to the add doctor section and fill the fields then Click on "Add doctor" button	successfully added
TC018	Adding doctor with missing or invalid information	A02	Login to the system then to the dashboard and navigate to the add doctor section and leave one or more field empty then Click on "Add doctor" button	error message of validations displayed with invalid input

8 Test Scenario 7

- **Admin' Actions and Behaviors:**
  The admin add other admin

- **Technical aspects of the requirement**

  - User Interface (UI): Design an intuitive and user-friendly interface for admins to add admin. Consider factors such as ease of use, accessibility, and responsiveness across different devices.
  - Database Management: Design a strong data model to store admin's information. The system should utilize a strong database management system to store data efficiently. Implement mechanisms such as constraints, validations, and referential integrity to prevent data corruption, duplication, or inconsistency.

8.1 Test Cases

Test Cases for the scenario mention in section 9 shown in Table 9

Table 8: Test Cases for Scenario 7

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC19	Successful addition of a new admin	A01	Login to the system then to the dashboard and navigate to the add doctor section and fill the fields then Click on "Add admin" button	successfully added
TC20	Adding admin with missing or invalid information	A01	Login to the system then to the dashboard and navigate to the add admin section and leave one or more field empty then Click on "Add admin" button	error message of validations displayed with invalid input

9 Test Scenario 8

- **Admin' Actions and Behaviors:**
  The admin updates his profile
- **Technical aspects of the requirement**
  - Database Management: The system should utilize a strong database management system to store admin profile data efficiently. Implement mechanisms such as constraints, validations, and referential integrity to prevent data corruption, duplication, or inconsistency.
  - User Authentication: Ensure that only authenticated admin users are allowed to access and update their profiles.
- **Possible Scenarios of System Failure**
  - Data Loss: The loss of admin profile's information can lead to disturbances in the operation of the clinic
  - Input Data Validation Failure:The system does not properly validate admin input data, leading to errors in the profile update process.
  - Software error and bugs: This error can cause unexpected actions, such as not editing the profile or data inconsistencies.

9.1 Test Cases

Test Cases for the scenario mention in section 9 shown in Table 9

Table 9: Test Cases for Scenario 8

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC21	Valid profile update	A07	Login to the system then to the profile page and navigate to the edit button and modify the fields and Click on "save" button	successfully updated
TC22	Cancel profile update	A07	Login to the system then to the profile page and navigate to the edit button and modify the fields and Click on "cancel" button	the profile remains the same without changes
TC23	Profile update with invalid input	A07	Login to the system then to the profile page and navigate to the edit button and modify the fields with invalid input and Click on "Save" button	Error message displayed