



Animal-Care Veterinary Clinic Management System

Part 3 - Project Development - Test
Plan

Lasalle College - Fall 2025

Web Server Application

Professor: Houria Hamel

Team: Negar Pirasteh - Betty Dang - Hope Jeanine
Ukundimana - Ngoc Yen Nhi Pham

Contents

Team Contribution Summary.....	Error! Bookmark not defined.
1. OBJECTIVES	2
2. TEST ENVIRONMENT	2
3. TEST DATA	2
4. FUNCTIONAL TEST CASES (BY CLASS / CONTROLLER)	2
4.1 Owners (OwnersController + Owner model)	2
4.2 Animals (AnimalsController + Animal model)	3
4.3 Veterinarians (VeterinariansController + Veterinarian model)	4
4.4 Schedules (SchedulesController + Schedule model)	5
4.5 Appointments (AppointmentsController + Appointment model)	5
4.6 VisitHistory (VisitHistoriesController + VisitHistory model)	6
4.7 UserAccount and Authentication (AccountController + UserAccount model)	7
4.8 Reports (ReportsController + MonthlyReportVM)	7
5. AUTHENTICATION AND AUTHORIZATION TESTS	8
6. VALIDATION AND UI TESTS	9
7. SUMMARY OF TEST RESULTS	9

1. OBJECTIVES

- Verify all CRUD operations for each database entity.
 - Verify that authentication and authorization work correctly for different roles.
 - Verify that form validation prevents invalid data.
 - Verify that monthly report calculations are correct.
 - Verify that the user interface behaves correctly and remains responsive.
-

2. TEST ENVIRONMENT

- **Operating System:** Windows 11
 - **Browser:** Google Chrome (latest)
 - **Server:** ASP.NET Core 8 (Kestrel / IIS Express)
 - **Database:** SQL Server 2022 + Animal-ClinicDB_T1
 - **Build Configuration:** Debug
-

3. TEST DATA

Sample records for:

- Owners (4 sample owners)
 - Animals (linked to each owner)
 - Veterinarians (2–3 vets)
 - Schedules (multiple dates/time slots per vet)
 - Appointments (booked, cancelled, completed)
 - VisitHistories (created after completed appointments)
 - UserAccounts (Admin, Secretary, Veterinarian users)
-

4. FUNCTIONAL TEST CASES (BY CLASS / CONTROLLER)

For each controller, we test **Insert, Update, Delete, Edit, List** as required.

4.1 Owners (OwnersController + Owner model)

1. Create Owner

- Steps: Go to Owners → Create. Fill all fields with valid data. Submit.
- Expected: New owner appears in Owners list.
- Result: PASS.

2. Edit Owner

- Steps: Owners → Edit first owner. Change address. Save.
- Expected: Updated address shown in list and details.
- Result: PASS.

3. Delete Owner

- Steps: Owners → Delete chosen owner → Confirm.
- Expected: Owner removed from list.
- Result: PASS.

4. List Owners

- Steps: Owners → Index.
- Expected: All owners displayed with full data.
- Result: PASS.

5. Details Owner

- Steps: Owners → Details.
- Expected: Detailed information for owner shown.
- Result: PASS.

4.2 Animals (AnimalsController + Animal model)

1. Create Animal

- Steps: Animals → Create. Choose an existing Owner, fill fields, submit.
- Expected: New animal appears in Animals list with correct owner.
- Result: PASS.

2. Edit Animal

- Steps: Animals → Edit → change Species and Age.

- Expected: Species and Age updated.
- Result: PASS.

3. Delete Animal

- Steps: Animals → Delete → Confirm.
- Expected: Animal removed from list.
- Result: PASS.

4. List Animals

- Steps: Animals → Index.
- Expected: List shows all animals with their owners.
- Result: PASS.

5. Validation

- Steps: Submit Create form with empty Name or invalid Age.
- Expected: Validation messages appear; record is not saved.
- Result: PASS.

4.3 Veterinarians (VeterinariansController + Veterinarian model)

1. Create Vet

- Expected: New veterinarian added; appears in list.
- Result: PASS.

2. Edit Vet

- Expected: Updated information is visible.
- Result: PASS.

3. Delete Vet

- Expected: Vet removed; any related schedules must be managed by business rules.
- Result: PASS.

4. List Vets / Details

- Expected: All vets displayed; details show full information.

- Result: PASS.
-

4.4 Schedules ([SchedulesController](#) + [Schedule model](#))

1. Create Schedule

- Steps: Schedules → Create. Select Vet, date, time slot, “Available”.
- Expected: New schedule appears in list.
- Result: PASS.

2. Prevent Conflicting Schedule

- Steps: Try to create second schedule for same vet, same date/time slot.
- Expected: Either validation error or business error message (no overlapping slot).
- Result: PASS (according to implemented rule).

3. Edit Schedule

- Expected: Updated date/time/status.
- Result: PASS.

4. Delete Schedule

- Expected: Schedule removed.
- Result: PASS.

5. List Schedules

- Expected: All schedules visible with veterinarian name.
 - Result: PASS.
-

4.5 Appointments ([AppointmentsController](#) + [Appointment model](#))

1. Create Appointment

- Steps: Appointments → Create. Select Animal and available Schedule, date/time, reason.
- Expected: Appointment created with Status “Booked”, appears in list.
- Result: PASS.

2. Edit Appointment

- Steps: Change Reason or Status.
- Expected: Updated values saved.
- Result: PASS.

3. **Cancel Appointment**

- Steps: Set Status to “Cancelled”.
- Expected: Status visible as cancelled in list.
- Result: PASS.

4. **Complete Appointment**

- Steps: Set Status to “Completed”.
- Expected: Can now create associated VisitHistory.
- Result: PASS.

5. **List Appointments**

- Expected: All appointments listed with animal name, date, time, status.
- Result: PASS.

4.6 VisitHistory (VisitHistoriesController + VisitHistory model)

1. **Create Visit History**

- Steps: VisitHistory → Create. Choose appointment and animal, fill date, diagnosis, treatment, prescription.
- Expected: New visit record appears in list.
- Result: PASS.

2. **Edit Visit History**

- Expected: Updated diagnosis/treatment saved.
- Result: PASS.

3. **Delete Visit History**

- Expected: Visit record removed.
- Result: PASS.

4. **List Visit History**

- Expected: All records shown with animal, appointment, and veterinarian information.
- Result: PASS.

5. Validation on Visit Date

- Steps: Submit form with invalid or missing VisitDate.
- Expected: Validation error, record not saved.
- Result: PASS.

4.7 UserAccount and Authentication (AccountController + UserAccount model)

1. Successful Login

- Steps: Enter correct credentials for Admin user.
- Expected: Redirect to Home/Index and navbar shows admin links.
- Result: PASS.

2. Failed Login

- Steps: Enter wrong password.
- Expected: Error message "Invalid username or password."
- Result: PASS.

3. Logout

- Steps: Click Logout.
- Expected: Auth cookie cleared, redirected to Login page.
- Result: PASS.

4. Role-based Menus

- Steps: Login as Admin, Secretary, Veterinarian.
- Expected: Navbar items change according to role (e.g., only Admin sees Reports).
- Result: PASS.

4.8 Reports (ReportsController + MonthlyReportVM)

1. Generate Current Month Report

- Steps: Login as Admin → Reports → Monthly (no parameters).
- Expected: Report for current month appears with totals and vet workload list.
- Result: PASS.

2. **Generate Specific Month**

- Steps: Pass year and month in query string (for example, 2025-11).
- Expected: Data filtered only for that month.
- Result: PASS.

3. **No Data Case**

- Steps: Choose a month with no records.
- Expected: Totals show zero and vet table shows no rows or “No data available.”
- Result: PASS.

5. AUTHENTICATION AND AUTHORIZATION TESTS

1. **Admin Access to All Controllers**

- Expected: Admin can open Animals, Owners, Veterinarians, Schedules, Appointments, VisitHistories, Reports.
- Result: PASS.

2. **Secretary Restrictions**

- Expected: Secretary can manage Owners, Animals, Appointments, Schedules but cannot access Reports or some admin-level functions.
- Result: PASS.

3. **Veterinarian Restrictions**

- Expected: Veterinarian can see appointments and manage VisitHistory but cannot create Owners or Animals.
- Result: PASS.

4. **Access Denied Page**

- Steps: Veterinarian tries to access Owners/Index manually.
- Expected: Redirected to AccessDenied page.

- Result: PASS.
-

6. VALIDATION AND UI TESTS

1. Required Field Validation

- For each Create/Edit form, try to submit empty required fields.
- Expected: Red validation messages next to each field.
- Result: PASS.

2. Format Validation

- Enter invalid email and phone formats.
- Expected: Validation messages appear.
- Result: PASS.

3. Date/Time Inputs

- Use invalid dates/times where possible.
- Expected: Browser prevents submit or server validation returns errors.
- Result: PASS.

4. Responsive UI

- Resize browser window, test on mobile width.
 - Expected: Navbar collapses correctly; forms remain usable.
 - Result: PASS.
-

7. SUMMARY OF TEST RESULTS

- All CRUD operations for **Owners, Animals, Veterinarians, Schedules, Appointments, VisitHistories** are working.
- Authentication and role-based authorization work correctly for **Admin, Secretary, and Veterinarian**.
- Server-side and client-side validations prevent invalid data entry.
- Monthly report feature correctly calculates totals and veterinarian workload by month.
- The system is stable and ready for final demonstration and evaluation.

