

GatoPointment: Cat Health Appointment System

Wan Ahmad Waafi Yasin Omar¹, Rozita Abdul Jalil^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: rozita@uthm.edu.my
DOI: <https://doi.org/10.30880/aitcs.2026.07.01.079>

Article Info

Received: 12 June 2025
Accepted: 18 June 2026
Available online: 30 June 2026

Keywords

GatoPointment, pet owner,
appointment booking, appointment
notification, veterinary clinic

Abstract

Managing veterinary appointments and cat health records is often challenging when done manually using paper forms or phone contact. This issues occurs at AnggunVet Batu Pahat's veterinary clinic, leading to missed appointments, incomplete health records, and inefficient communication between pet owners and veterinary clinics. This project introduces a web-based appointment system named GatoPointment for AnggunVet Batu Pahat. The system is designed to help pet owners book appointments, view their cats' health details, and receive email for appointment confirmation or reminders for upcoming appointments. It also allows the veterinary clinic to manage their schedules efficiently and access shared health record details for pet owners. The system will be developed using PHP, MySQL, and HTML/CSS, with email notifications powered by PHPMailer to inform users about appointment approvals and upcoming appointment reminders. The project followed a structured design and testing process to ensure the system is easy to use and reliable. Future improvements could include adding a mobile app or advanced features like automatic health tracking to make the system even more helpful for users and the clinic.

1. Introduction

The current method of health appointment in AnggunVet Batu Pahat relies on manual processes. Pet owners often record basic health details, like vaccination dates and health concerns, either on paper or via spreadsheets. People increasingly need new technologies to make their daily lives easier and simpler [2]. Most pet owners manually manage their cat's healthcare, which often results in missed appointments or delays in treatment.

Currently, the approach to managing cat health primarily depends on periodic check-ups, scheduled vaccinations, and symptom reporting during in-person visits or messaging. On the other hand, customer have demanded digitally planned, organized, and timely managed services. This changing demand-supply connection between veterinary clinics and their clients has allowed advancements in primary online services[2]. GatoPointment aims to resolve these issues by offering a web-based platform for improved appointment scheduling.

2. Related Work

A review of the existing literature and related systems to the development of the GatoPointment: Cat Health Appointment System is conducted.

2.1 Case Study

Pet owners often face difficulties when scheduling, managing, and keeping track of veterinary appointments for their cats. It can lead to missed or delayed visits, impacting their pets' overall health and well-being. Without an organized system, important veterinary appointments, such as vaccinations or check-ups, may be forgotten or postponed, increasing the risk of health complications for cats.

2.2 Method or Technology

Developd GatoPointment Cat Health Appointment System using modern web development technologies and data management techniques to create an efficient, and user-friendly platform. GatoPointment uses modern web development technologies such as HTML, CSS, JavaScript, Node.js, and MySQL. These technologies are selected based on their reliability, efficiency, and suitability for building a web-based application.

2.3 Study of Existing Related Systems

Reviewing of existing pet health management and appointment booking systems provides valuable insights for the developing the GatoPointment. This section examines similar systems to identify best practices, strengths, and limitations, helping shape the design and features of the new system. The comparison below summarizes the key features of the three existing systems and the proposed GatoPointment. While each system includes appointment booking, reminders, and scheduling functionalities, the proposed system also includes health record management, distinguishing it from other systems.

Table 1 *System's Comparison*

Features/System	Qmed.asia	Vetstoria	KumoVet	GatoPointment
User Log	√	√	√	√
Online Appointment Booking	√	√	√	√
Automated Reminders & Notifications	√	√	√	√
Real-Time Availability	X	√	√	√
Health Record Management	X	X	√	√
Scheduling Optimization	X	√	√	√
Follow-up Appointment	X	X	X	√

3. Methodology/Framework

Agile methodology is ideal for developing the GatoPointment due to its flexibility, collaborative approach, and focus on delivering functional components in iterations. Agile allows close collaboration between the development team and stakeholders, including pet owners and veterinary clinic staff, throughout the development process. Stakeholder feedback is incorporated in every sprint review, ensuring the system addresses real-world challenges, such as streamlining appointment scheduling and health record management.

Table 2 *Software development activities and their task*

Phase	Task	Output
Planning	Propose the project to AnggunVet clinic, gather project requirements, and schedule activities and tasks.	Project proposal, requirement document, and task list
Design	Create wireframes and prototype, design database schema, and plan system architecture.	Wireframe design, UI design, database schema, system architecture.
Development	Build the front-end using HTML, CSS and JavaScript, set back-end with Node.js, integrate MySQL database	Functional web-based system, back-end API
Testing	Conduct unit testing, conduct usability testing, fix identified issues or bugs	Test result, updated system code, bug reports
Deployment	Host the system on server, connect database to live server, conduct final checks	Deployed application, deployment checklist

Review	Present the system to AnggunVet for review and collect data, analyse feedback and identify improvement areas	Feedback report, system update plan
--------	--	-------------------------------------

4. System Analysis and Design

4.1 System Requirements

Analyse and define the requirements for the GatoPointment. The system requirement analysis is a critical phase in the software development process as it identifies and documents the functional and non-functional needs of the system. This ensures that the system is designed to meet the expectations of both the pet owners and the veterinary clinic staff.

Table 3 *Functional Requirements*

No	Requirement	Description
1	User Registration and Login	Allow pet owners to create accounts and securely log in.
2	Appointment Scheduling	Enable pet owners to book, reschedule, or cancel appointments with veterinary clinics.
3	Health Record Management	Allow pet owners to log and view cat health data.
4	Dashboard for Clinic Staff	Provide clinic staff with access to manage appointments, view shared health data, and organize their schedules efficiently.
5	Report Module	Staff can generate report for analysis insight.

Table 4 *Non-Functional Requirements*

No	Requirement	Description
1	Performance	The system should handle user requests, such as booking or retrieving health records.
2	Operational	The system should be available 24/7.
3	Security	All sensitive information, such as pet health records and payment details, must be encrypted both in transit (using HTTPS) and at rest (using database encryption).
4	Cultural and political	Cultural sensitivities, such as appropriate language or symbols, should be considered in the design of the user interface.

Table 5 *User Requirements*

No	Requirement
1.	All users must have an account with a valid ID and password to access the system.
2.	Pet owners should be able to book, modify, and cancel appointments.
3.	Pet owners should be able to access and view their pet's health records.
4.	Pet owners should receive notifications when the appointment request approve or decline.
5.	Staff should be able to view and manage appointments.
6.	Veterinarians should be able to access and update pet health records.
7.	Staff should be able to generate and review reports on pet health and appointment statistics.
8.	Staff should be able to generate reports on system usage, pet health trends, and appointment statistics.
9.	Administrators should be able to create, modify, and delete user accounts (pet owners and veterinary staff).

4.2 System Design

The system analysis results, which is a data flow diagram and entity relationship diagram, are defined in this sub-section. A systematic approach is used to perform system analysis.

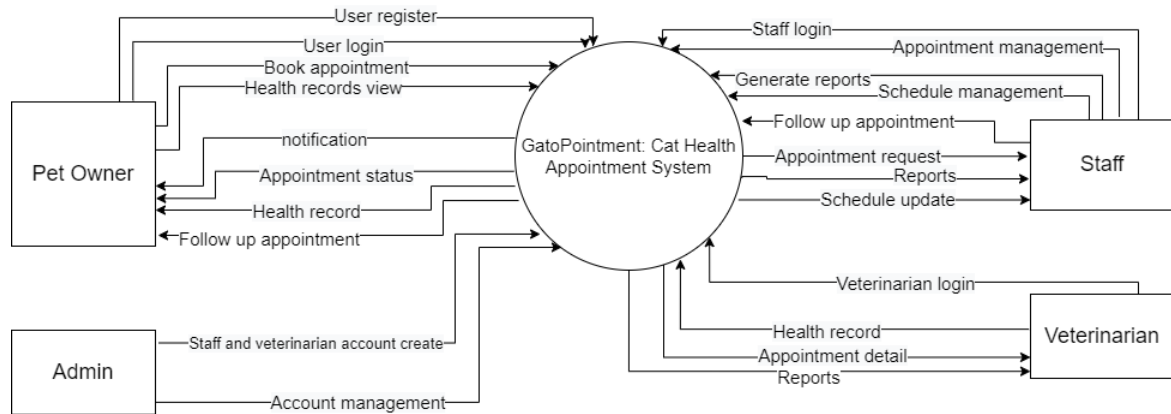


Fig. 1 Context Diagram

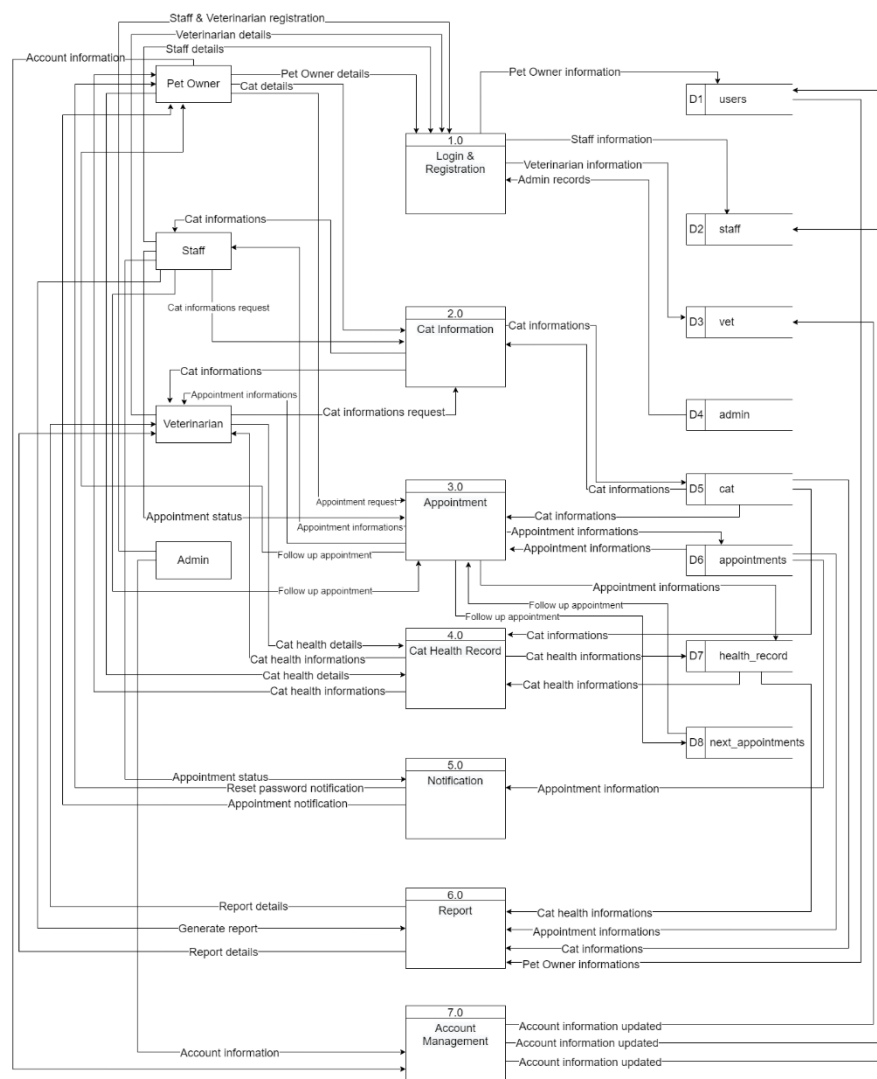


Fig. 2 Data Flow Diagram Level 0

An Entity Relationship Diagram (ERD) is a visual representation of how various entities within a system are connected, ensuring structural integrity and proper database design. In this system, seven primary tables have been identified: Users, Staff, Veterinarian, Admin, Cat, Appointment, and Health Record.

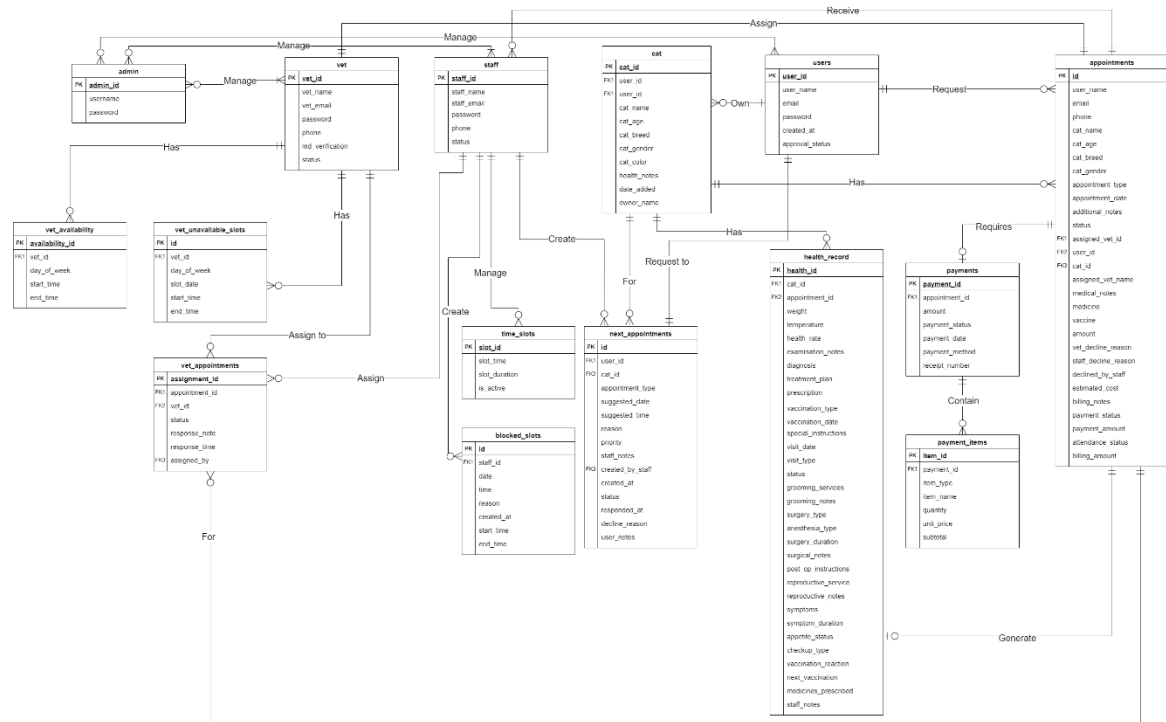


Fig. 3 Entity Relationship

4.3 System Design

After all the users' requirements are successfully analyzed, the project will proceed in design phase. In this phase, both interface and database had been designed in order to help visualize the system before proceeding with the coding the system.

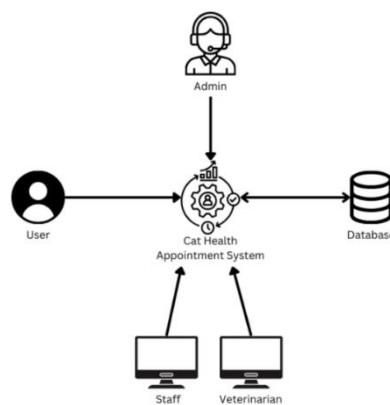


Fig. 4 System Architecture

A flowchart is a diagram that shows the steps in a process. It's used to visualize, analyze, and communicate processes. Figure 4 shows the comprehensive flowchart for the GatoPointment. It illustrates the process flows for all user roles including pet owners, staff, veterinarians, and administrators

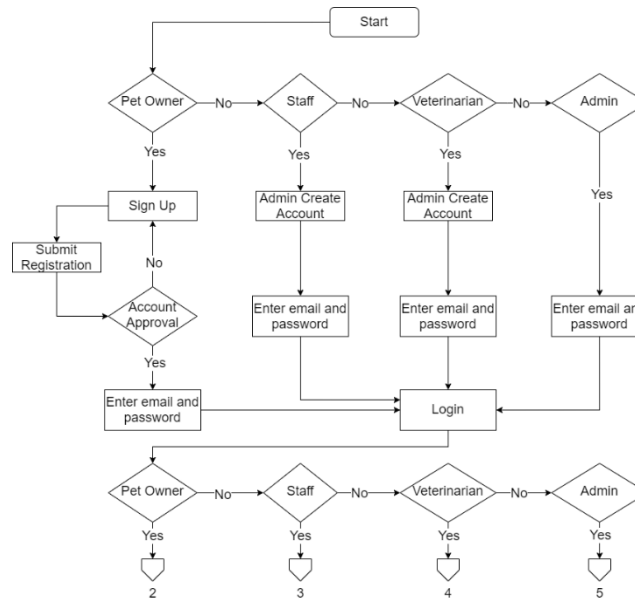


Fig. 5 System Flowchart Login

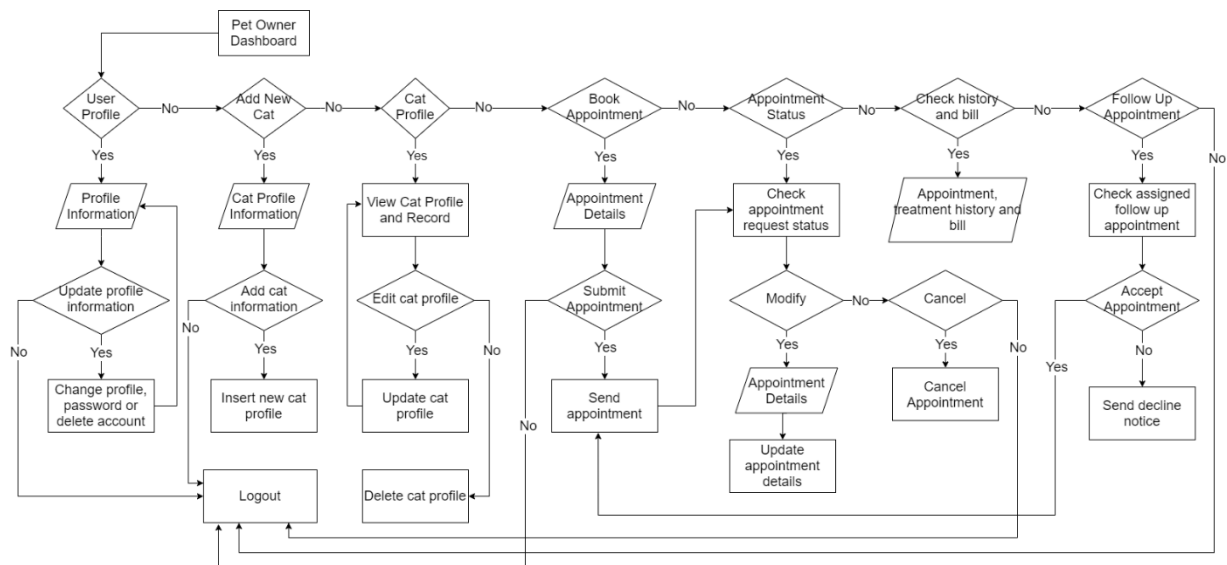


Fig. 6 System Flowchart Pet Owner

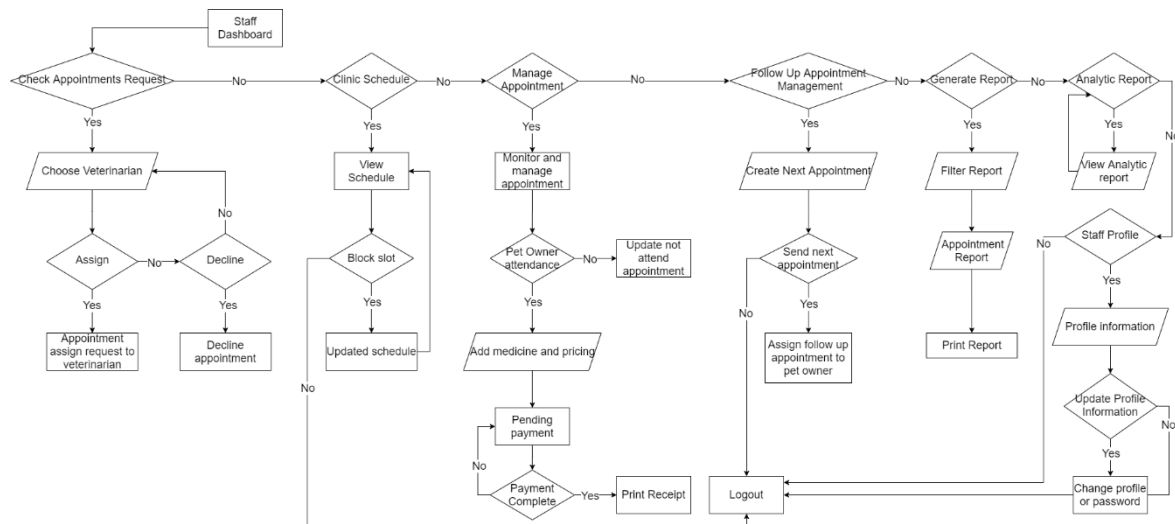


Fig. 7 System Flowchart Staff

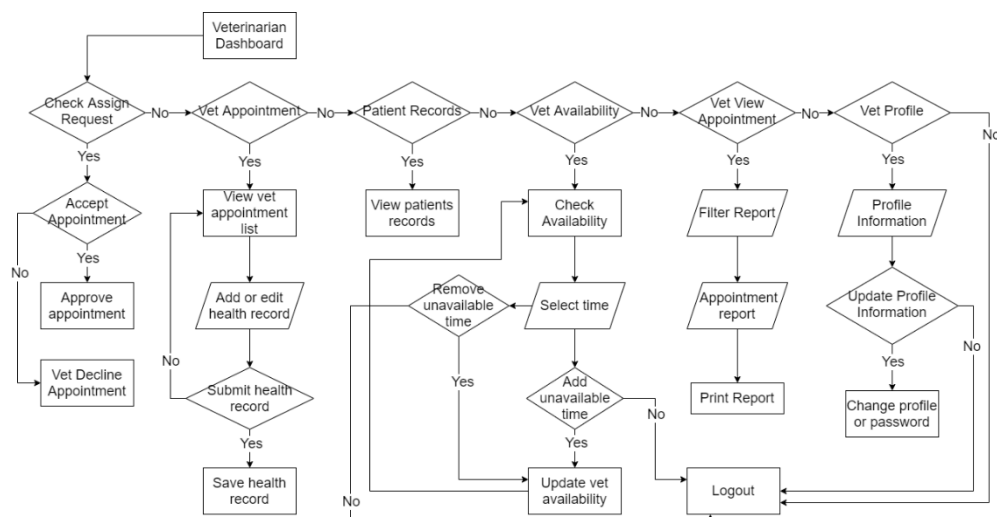


Fig. 8 System Flowchart Veterinarian

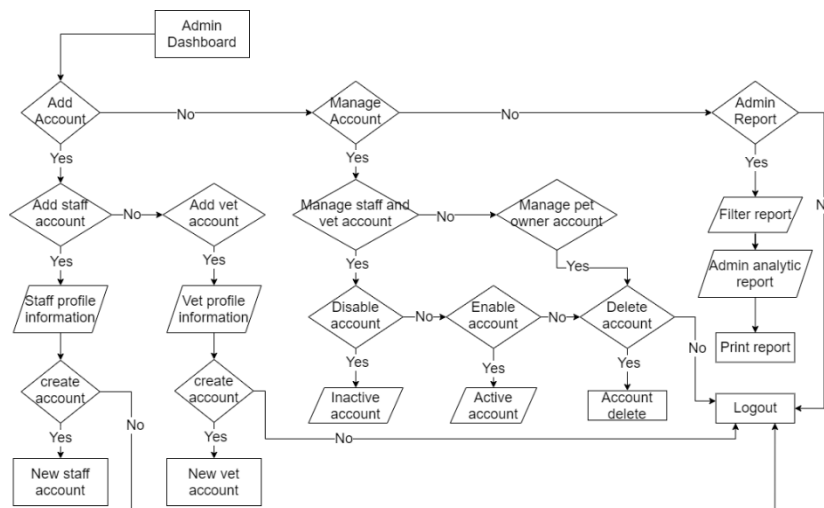
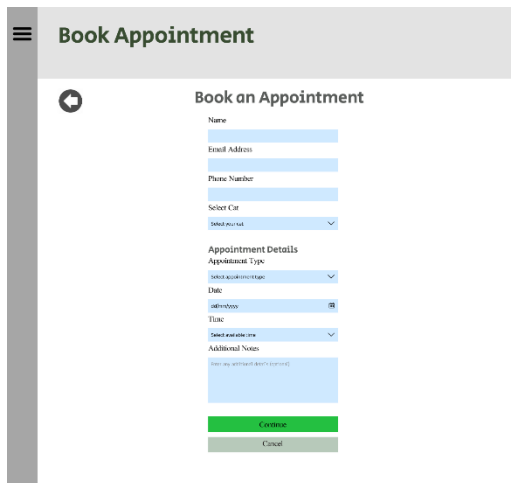


Fig. 9 System Flowchart Admin

4.4 Interface Design

The interface design focuses on user-friendly, simplicity, and functionality to interactions for pet owners, clinic staff, veterinarians, and administrators.



Book Appointment

Book an Appointment

Name

Email Address

Phone Number

Select Cat

Appointment Details

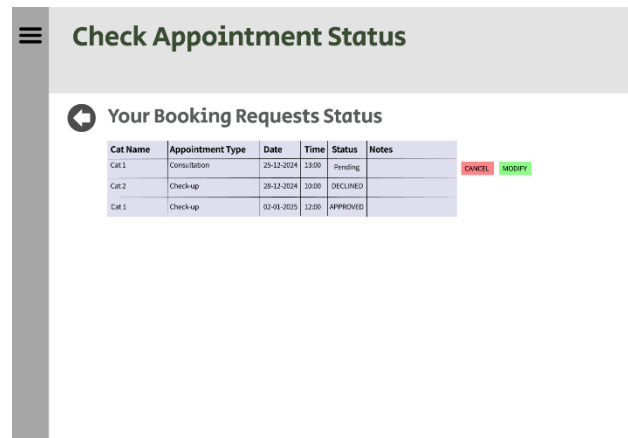
Appointment Type

Date

Time

Additional Notes

Fig. 10 Book Appointment Form

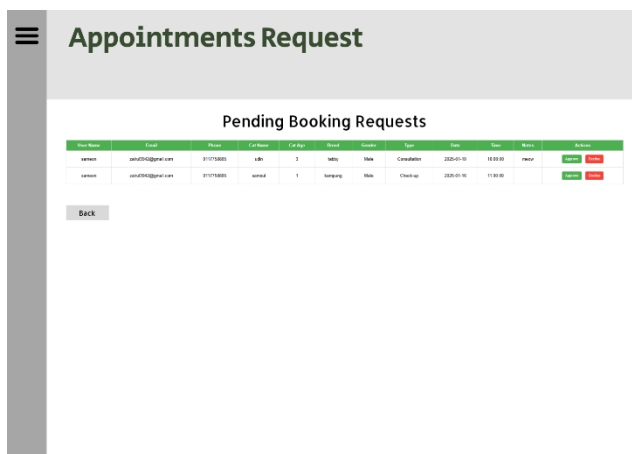


Check Appointment Status

Your Booking Requests Status

Cat Name	Appointment Type	Date	Time	Status	Notes
Cat 1	Consultation	25-12-2024	13:00	Pending	
Cat 2	Check-up	28-12-2024	10:00	DECLINED	
Cat 1	Check-up	02-01-2025	12:00	APPROVED	

Fig. 11 Check Appointment Status Form

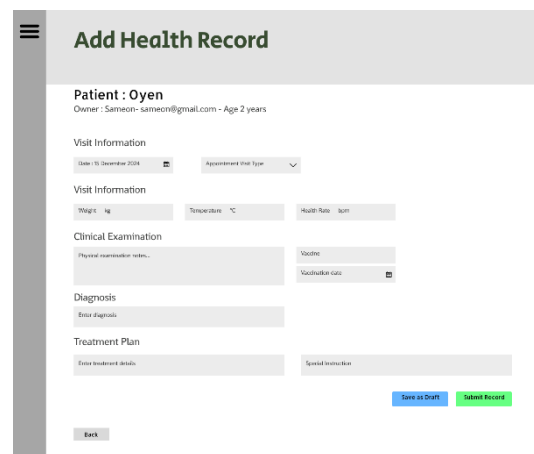


Appointments Request

Pending Booking Requests

Owner Name	Email	Phone	Cat Name	Cat Age	Gender	Type	Date	Time	Status	Action
sameon	sameon52@gmail.com	011718085	Tabby	2	Male	Consultation	2024-12-10	10:00:00	Pending	Cancel Approve
sameon	sameon52@gmail.com	011718085	sameon	1	sameon	Check-up	2024-12-10	11:00:00	Pending	Cancel Approve

Fig. 12 Appointment Request Form



Add Health Record

Patient : Oyen
Owner : Sameon - sameon52@gmail.com - Age 2 years

Visit Information

Last visit Appointment Date Type

Visit Information

Weight kg Temperature °C Health Note

Clinical Examination

Physical examination notes... Vaccine

Vaccination date

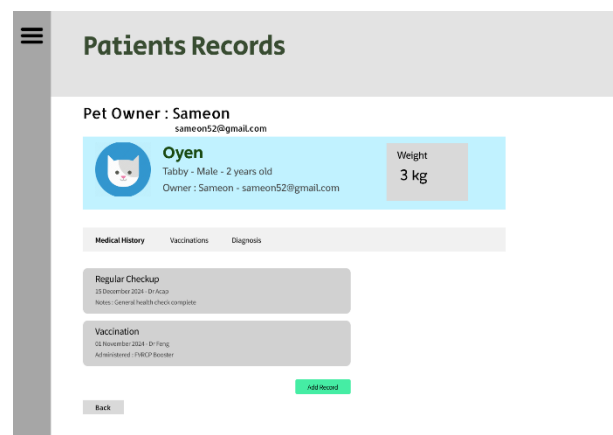
Diagnosis

Enter diagnosis

Treatment Plan

Enter treatment details Special instruction

Fig. 13 Add Health Record Form



Patients Records

Pet Owner : Sameon
sameon52@gmail.com

Oyen
Tabby - Male - 2 years old
Owner : Sameon - sameon52@gmail.com

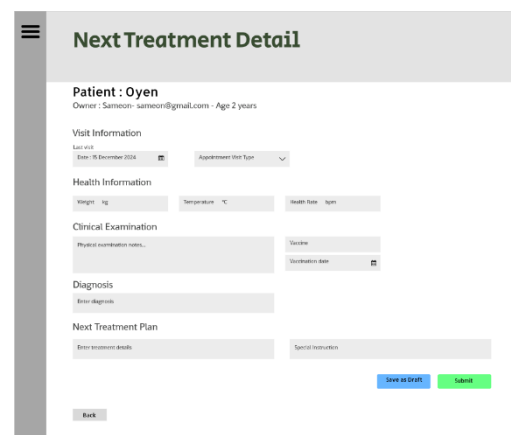
Weight **3 kg**

Medical History Vaccinations Diagnosis

Regular Checkup
15 December 2024 - Dr. Acas
Notes: General health check complete.

Vaccination
05 November 2024 - Dr. Hong
Administered: FIVCP Booster

Fig. 14 Patient Record Page



Next Treatment Detail

Patient : Oyen
Owner : Sameon - sameon52@gmail.com - Age 2 years

Visit Information

Last visit Appointment Date Type

Visit Information

Weight kg Temperature °C Health Note

Clinical Examination

Physical examination notes... Vaccine

Vaccination date

Diagnosis

Enter diagnosis

Next Treatment Plan

Enter treatment details Special instruction

Fig. 15 Check Next Treatment Page

5. Result and Discussion

This section focuses on system implementation and testing for the web-based GatoPointment: Cat Health Appointment System. It covers the implementation of all features, including login and registration, user management, booking management, schedule management, health record management and reporting.

5.1 Implementation

This section describes the development of functional modules in a system.

5.1.1 Login & Registration Interface

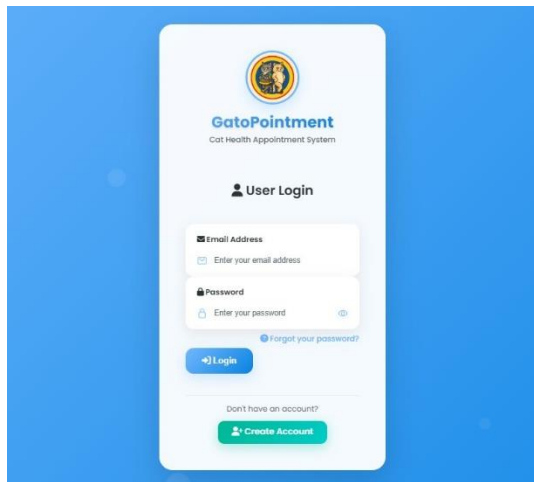


Fig. 16 Pet owner login interface

```
$email = $_POST['email'];
$password = $_POST['password'];
$name = $_POST['user_name'];

$stmt = $conn->prepare("SELECT * FROM users WHERE email = ?");
$stmt->bind_param("s", $email);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();

    if (password_verify($password, $row['password'])) {
        $_SESSION['user_id'] = $row['user_id'];
        $_SESSION['email'] = $row['email'];

        header("Location: dashboard.php");
        exit();
    }
}
```

Fig. 17 Pet owner login source code

Figure 16 show the login interface for pet owner. In this process, pet owners access the system using their email address and password. The interface includes registration options for new users. Upon successful validation, pet owners are directed to their dashboard. Failed login attempts trigger error messages and prompt credential re-entry.

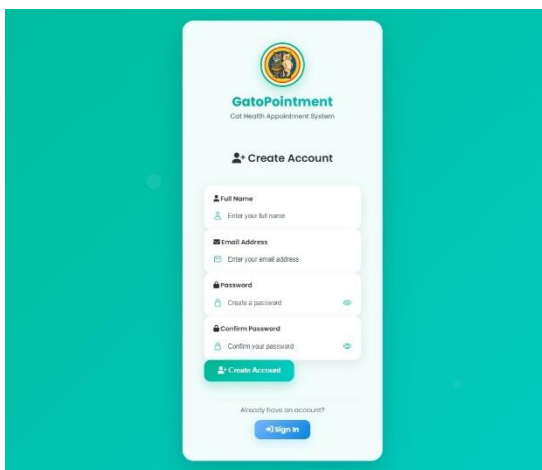


Fig. 18 Pet owner register interface

```
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $name = $conn->real_escape_string($_POST['user_name']);
    $email = $conn->real_escape_string($_POST['email']);
    $password = $conn->real_escape_string($_POST['password']);
    $confirm_password = $conn->real_escape_string($_POST['confirm_password']);

    if ($password !== $confirm_password) {
        echo "<script>alert('Error: Passwords do not match.');"; window.location.href = 'register.html'; exit();
    }

    $hashed_password = password_hash($password, PASSWORD_DEFAULT);

    $sql = "INSERT INTO users (user_name, email, password) VALUES ('$name', '$email', '$hashed_password')";

    if ($conn->query($sql) === TRUE) {
        echo "<script>alert('Registration successful! Redirecting to login page...');"; window.location.href = 'login.html';
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}
```

Fig. 19 Pet owner register source code

The registration form in Figure 18 allows new users to self-register through a public interface. Users must provide their name, email address, and password with confirmation. The system validates all entries in real-time, ensuring proper email formatting, password requirements, and matching password confirmation fields. Upon successful validation and submission via the "Create Account" button.

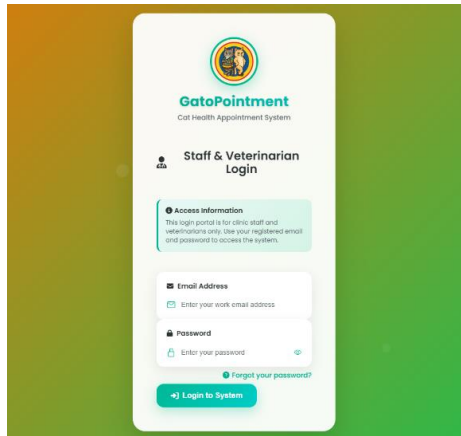


Fig. 20 Staff and veterinarian login interface

```
// Find user in vet table from
$stmt_query = "SELECT * FROM vet WHERE vet_email = ?";
$stmt_exec = $conn->prepare($stmt_query);
$stmt_bind_param($stmt_exec, "s", $email);
$stmt_execute($stmt_exec);
$stmt_result = $stmt_exec->get_result();

if ($stmt_result && $stmt_result->num_rows > 0) {
    $stmt = $stmt_result->fetch_assoc();

    if (password_verify($password, $stmt['password'])) {
        $token = bin2hex(random_bytes(32));

        $stmt_id = $stmt['vet_id'];
        $stmt_data = [
            'vet_id' => $stmt_id,
            'vet_email' => $stmt['vet_email'],
            'token' => $token,
            'logged_in' => true,
            'login_time' => time()
        ];

        $session_dir = 'sessions';
        if (!is_dir($session_dir)) {
            mkdir($session_dir, 0750, true);
        }

        $session_file = $session_dir . '/' . $stmt_id . '.json';
        file_put_contents($session_file, json_encode($stmt_data));

        setcookie('vet_token', $token, time() + 86400, '/');

        header("Location: dashboard_vet.php");
        exit();
    } else {
        echo "<script>alert('Invalid password for veterinarian account.');

```

Fig. 21 Staff and veterinarian login source code

Figure 20 show the staff login interface provides access points for clinic staff and veterinarian. Staff members or veterinarians enter their designated email and password. Upon successful authentication, they access the designated management dashboard based on which email categories available in system databases, whether it for staff or veterinarians. Invalid credentials trigger error notifications requesting re-entry.

5.1.2 Appointment Interface

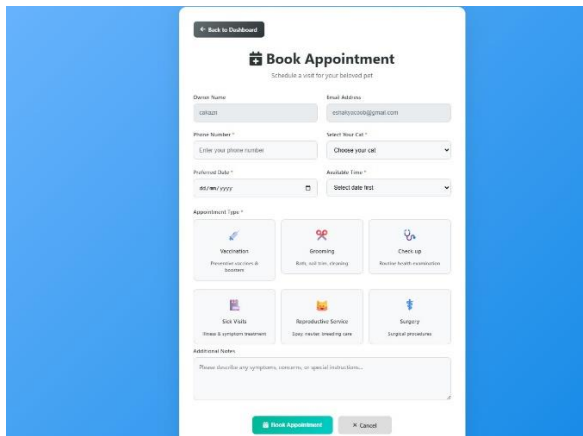


Fig. 22 Pet owner booking interface

```
// Insert appointment
$insert_query = "INSERT INTO appointments (
    user_id, cat_id, user_name, email, phone,
    cat_name, cat_age, cat_breed, cat_gender,
    appointment_type, appointment_date, appointment_time,
    additional_notes, status
) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, 'Pending')";

$stmt = $conn->prepare($insert_query);
$stmt->bind_param(
    "iisssssssss",
    $user_id, $cat_id, $cat_data['user_name'], $cat_data['email'],
    $phone, $cat_data['cat_name'], $cat_data['cat_age'],
    $cat_data['cat_breed'], $cat_data['cat_gender'], $appointment_type,
    $appointment_date, $appointment_time, $additional_notes
);

if (!$stmt->execute()) {
    echo "<script>alert('Error booking appointment: " . $stmt->error . "');

```

Fig. 23 Pet Owner booking source code

Figure 22 show pet owner's appointment booking interface provides a form for pet owner to fill out all important information for the booking process. Pet owner can click on the book appointment for submitting the appointment and will send to staff.

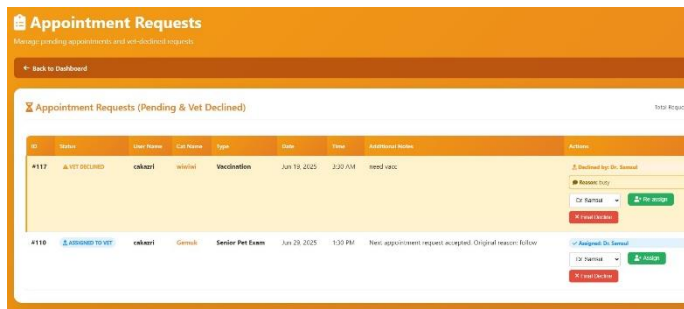


Fig. 24 Staff assign appointment interface

Figure 24 show staff can view pending appointments booking requested by pet owner in table format. The table displays essential appointment information. Each request features action buttons to assign appointment to veterinarian which first staff can choose veterinarian to assign and will display assigned to selected veterinarian. If the assigned veterinarian declined the task, the status would update that vet declined, and staff can re-assign to other veterinarian or staff can make a final decline which will decline the appointment to pet owner.

```
// Handle design to vet decline
if ($SERVER['REQUEST_METHOD'] == "POST" && isset($_POST['assign_vet'])) {
    $appointment_id = intval($_POST['appointment_id']);
    $vet_id = intval($_POST['vet_id']);

    // Get vet details:
    $vet_query = "SELECT vet_id, vet_name, vet_email FROM vet WHERE vet_id = ?";
    $vet_stat = $conn->prepare($vet_query);
    $vet_stat->bind_param("i", $vet_id);
    $vet_stat->execute();
    $vet_result = $vet_stat->get_result();

    if ($vet_result->num_rows == 0) {
        echo "<script>alert('Veterinarian not found.');

```

Fig. 25 Staff assign appointment code



Fig. 26 Vet appointment request interface

Veterinarian's check appointment request interface in Figure 26 important for vet to approve or decline the appointment that assigned by staff. Veterinarian can approve or decline the appointment. If veterinarian approve the appointment, the appointment will update in the system database and email notification will send to pet owner meanwhile if veterinarian decline the appointment, the popup will appear for veterinarian to state the reason for declining and notify the staff first for staff to determine the declined appointment.

```
// Verify appointment is assigned to selected vet
$check_query = "SELECT * FROM appointments WHERE id = ? AND assigned_vet_id = ? AND status = 'Pending'";
$check_stat = $conn->prepare($check_query);
$check_stat->bind_param("ii", $appointment_id, $vet_id);
$check_stat->execute();
$check_result = $check_stat->get_result();

if ($check_result->num_rows == 0) {
    echo "<script>alert('This appointment is not assigned to you or is no longer pending.');

```

Fig. 27 Vet appointment request code

5.1.3 Notification Interface

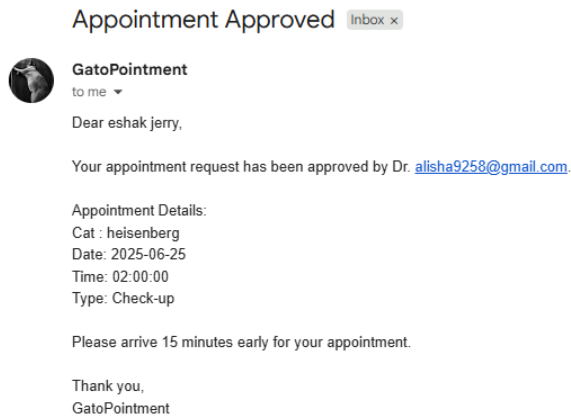


Fig. 28 Pet Owner Appointment Approved email

```
// Send email notification to user
try {
    $mail = new PHPMailer(true);

    // Server settings
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'wanahmaduaf120@gmail.com'; // Email address
    $mail->Password = 'krzg jbhf sybi wywo'; // App Password
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;

    // Check if email is valid
    if (filter_var($user_email, FILTER_VALIDATE_EMAIL)) {
        echo "script>alert('Booking Approved but email address is invalid.');
```

Fig. 29 Pet Owner Appointment Approve Code

Figure 28 shows an email notification receive by pet owner email address, the email context notified pet owner about appointment status approval and information about assigned veterinarian, appointment details. Figure 29 shows the PHPMailer coding process of sending the appointments approve notification.

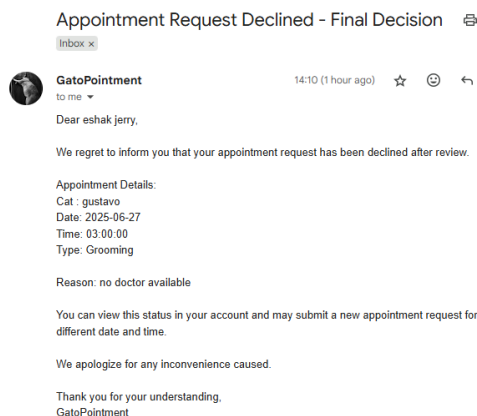


Fig. 30 Pet Owner Appointment Declined email

```
// Send email notification to user
try {
    $mail = new PHPMailer(true);

    // Server settings
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'wanahmaduaf120@gmail.com';
    $mail->Password = 'krzg jbhf sybi wywo';
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;

    // Check if email is valid
    if (filter_var($appointment['email'], FILTER_VALIDATE_EMAIL)) {
        // Email settings
        $mail->setFrom('wanahmaduaf120@gmail.com', 'GatoPointment');
        $mail->addAddress($appointment['email'], $appointment['user_name']);
        $mail->Subject = 'Appointment Request Declined - Final Decision';
        $mail->Body = 'Dear ' . $appointment['user_name'] . ',\n\n';
        $mail->Body .= 'We regret to inform you that your appointment request has been de';
        $mail->Body .= 'Appointment Details:\n\n';
        $mail->Body .= 'Cat : ' . $appointment['cat_name'] . '\n\n';
        $mail->Body .= 'Date: ' . $appointment['appointment_date'] . '\n\n';
        $mail->Body .= 'Time: ' . $appointment['appointment_time'] . '\n\n';
        $mail->Body .= 'Type: ' . $appointment['appointment_type'] . '\n\n';

        if (empty($decline_reason)) {
            $mail->Body .= 'Reason: ' . $decline_reason . '\n\n';
        }

        $mail->Body .= 'You can view this status in your account and may submit a new appo';
        $mail->Body .= 'We apologize for any inconvenience caused.\n\n';
        $mail->Body .= 'Thank you for your understanding, GatoPointment';

        $mail->send();
    }
} catch (Exception $e) {
    echo "script>alert('Appointment assigned to ' . $vet['vet_name'] . ') but failed to send email: ' . $e->getMessage());';
    exit();
}
```

Fig. 31 Pet Owner Appointment Decline Code

Figure 30 shows an email notification receive by pet owner email address, the email context notified pet owner about appointment status decline and information about declined appointment details with reason of decline. Figure 31 shows the PHPMailer coding process of sending the appointments decline notification.

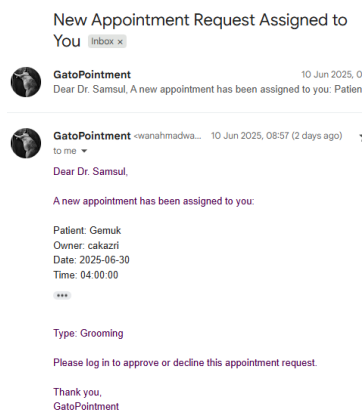


Fig. 32 Veterinarian Appointment Assign email

```
try {
    $mail = new PHPMailer(true);

    // Server settings
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'wanahmaduaf120@gmail.com';
    $mail->Password = 'krzg jbhf sybi wywo';
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;

    // Email settings
    $mail->setFrom('wanahmaduaf120@gmail.com', 'GatoPointment');
    $mail->addAddress($vet['vet_email'], $vet['vet_name']);
    $mail->Subject = 'New Appointment Request Assigned to You';
    $mail->Body = 'Dear Dr. ' . $vet['vet_name'] . ',\n\n';
    $mail->Body .= 'A new appointment has been assigned to you:\n\n';
    $mail->Body .= 'Cat : ' . $appointment['cat_name'] . '\n\n';
    $mail->Body .= 'Owner: ' . $appointment['user_name'] . '\n\n';
    $mail->Body .= 'Date: ' . $appointment['appointment_date'] . '\n\n';
    $mail->Body .= 'Time: ' . $appointment['appointment_time'] . '\n\n';
    $mail->Body .= 'Type: ' . $appointment['appointment_type'] . '\n\n';
    $mail->Body .= 'Please log in to approve or decline this appointment request.\n\n';
    $mail->Body .= 'Thank you, GatoPointment';

    $mail->send();
    echo "script>alert('Appointment assigned to ' . $vet['vet_name'] . ') and notification sent successfully!'; window.
    exit();
} catch (Exception $e) {
    echo "script>alert('Appointment assigned to ' . $vet['vet_name'] . ') but failed to send email: ' . $e->getMessage());';
    exit();
}
```

Fig. 33 Veterinarian Appointment Assign Code

Figure 32 shows an email notification received by veterinarian email address, the email context notified the new appointment request assigned. Information about appointment details display for veterinarian to alert the type of

appointment. Figure 33 shows the PHPMailer coding process of sending the appointments assign request notification.

5.1.4 Health Record Interface

Fig. 34 Vet medical record form interface

```
$weight = isset($_POST['weight']) && !empty($_POST['weight']) ? floatval($_POST['weight']) : null;
$temperature = isset($_POST['temperature']) && !empty($_POST['temperature']) ? floatval($_POST['temperature']) : null;
$health_rate = isset($_POST['health_rate']) && !empty($_POST['health_rate']) ? intval($_POST['health_rate']) : null;
$examination = cleanformdata($_POST['examination']) ?? '';
$diagnosis = cleanformdata($_POST['diagnosis']) ?? '';
$treatment = cleanformdata($_POST['treatment']) ?? '';
$prescription = cleanformdata($_POST['prescription']) ?? '';
$notes = cleanformdata($_POST['notes']) ?? '';
$vaccination = cleanformdata($_POST['vaccination']) ?? '';
$vaccination_date = isset($_POST['vaccination_date']) && !empty($_POST['vaccination_date']) ? $_POST['vaccination_date'] : null;
$status = isset($_POST['save_draft']) ? 'draft' : 'submitted';

// Get cat_id from appointment
$cat_id = $appointment['cat_id'];

// Get visit_date and visit_type
$visit_date = date('Y-m-d');
$visit_type = $appointment['appointment_type'] ?? 'General';

// Initialize all additional fields as null
$grooming_services = null;
$grooming_notes = null;
$surgery_type = null;
$anesthesia_type = null;
$surgery_duration = null;
$surgical_notes = null;
$spont_op_instructions = null;
$reproductive_service = null;
$reproductive_notes = null;
$symptoms = null;
$symptoms_status = null;
$apoptosis_status = null;
$checkup_type = null;
$vaccination_reaction = null;
$next_vaccination = null;

// Get fields based on appointment type
$appointment_type = strtolower($appointment['appointment_type']) ?? 'check-up';

if (strpos($appointment_type, 'vaccination') !== false) {
    $vaccination_reaction = cleanformdata($_POST['vaccination_reaction']) ?? '';
    $next_vaccination = isset($_POST['next_vaccination']) && !empty($_POST['next_vaccination']) ? $_POST['next_vaccination'] : null;
} else {
    $grooming_services = isset($_POST['grooming_services']) && !empty($_POST['grooming_services']) ? implode(', ', $_POST['grooming_services']) : null;
}
```

Fig. 35 Vet medical record form code

Figure 34 interface is important for veterinarian to key in the medical record for each appointment visit. The medical record form will be different based on appointment type that suitable for veterinarian to key in the information form.

Fig. 36 Vet view patient record interface

```
<?php
$owner_query = "SELECT user_name, email FROM users WHERE user_id = ?";
$owner_stmt = $conn->prepare($owner_query);
$owner_stmt->bind_param("i", $owner_id);
$owner_stmt->execute();
$owner_result = $owner_stmt->get_result();
$owner_info = $owner_result->fetch_assoc();

$scats_query = "SELECT c.cat_id, c.cat_name, c.cat_age, c.cat_breed, c.cat_gender, c.cat_color, c.date_added,
COUNT(hr.health_id) as records_count,
MAX(hr.visit_date) as last_visit
FROM cat c
LEFT JOIN health_record hr ON c.cat_id = hr.cat_id
WHERE c.user_id = ?
GROUP BY c.cat_id, c.cat_name, c.cat_age, c.cat_breed, c.cat_gender, c.cat_color, c.date_added
ORDER BY c.cat_name ASC";

$scats_stmt = $conn->prepare($scats_query);
$scats_stmt->bind_param("i", $owner_id);
$scats_stmt->execute();
$scats_result = $scats_stmt->get_result();
$scats = $scats_result->fetch_assoc();
>>
```

Fig. 37 Vet view patient record code

Figure 36 shows full detailed medical records that will be shown for the veterinarian to view. The medical records will show all the details keyed in by veterinarian and staff during visits that are based on appointment type.

Fig. 38 Pet owner view cat record interface

```
$cat = $result->fetch_assoc();

// Fetch appointment history
$appointment_query = "SELECT hr.* FROM appointments WHERE cat_id = ? ORDER BY appointment_date DESC, appointment_time DESC";
$appointment_stmt = $conn->prepare($appointment_query);
$appointment_stmt->bind_param("i", $cat_id);
$appointment_stmt->execute();
$appointment_result = $appointment_stmt->get_result();

// Fetch health records
$health_query = "SELECT hr.* FROM health_record hr
JOIN appointments a ON hr.appointment_id = a.id
WHERE hr.cat_id = ?
ORDER BY a.appointment_date DESC";
$health_stmt = $conn->prepare($health_query);
$health_stmt->bind_param("i", $cat_id);
$health_stmt->execute();
$health_result = $health_stmt->get_result();
```

Fig. 39 Pet owner view cat record code

Figure 38 shows the interface through which the pet owner can view the entire history of the cat's medical record information. This would ease the pet owner's tracking of the cat's health treatment.

5.1.5 Report Interface

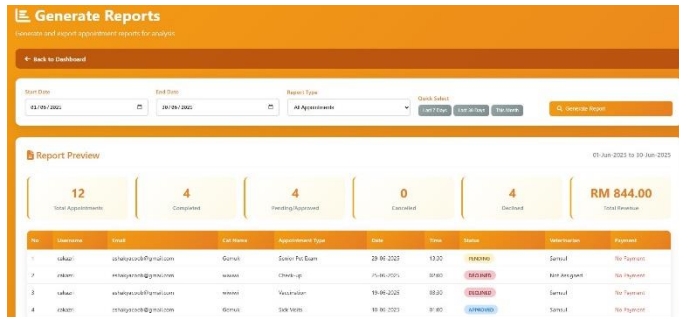


Fig. 40 Staff generate report interface

```

$start_date = $GET['start_date'] // date('Y-m-d');
$end_date = $GET['end_date'] // date('Y-m-d');
$report_type = $GET['report_type'] // 'all';

// if ($report_type != 'all') {
//     $start_date = date('Y-m-d');
//     $end_date = date('Y-m-d');
// }

if (strtotime($start_date) > strtotime($end_date)) {
    $temp = $start_date;
    $start_date = $end_date;
    $end_date = $temp;
}

$where_condition = "WHERE a.appointment_date BETWEEN ? AND ?";
$params = [$start_date, $end_date];
$param_types = "is";

if ($report_type != 'all') {
    $where_condition .= " AND a.status = ?";
    $params[] = $report_type;
    $param_types .= "s";
}

$appointments_query = "
SELECT
    a.id,
    a.user_name,
    a.email,
    a.cat_name,
    a.appointment_type,
    a.appointment_date,
    a.appointment_time,
    a.status,
    a.created_at,
    a.assigned_vet_name,
    a.attendance_status,
    p.amount as payment_amount,
    p.payment_status
FROM appointments a
LEFT JOIN payments p ON a.id = p.appointment_id
$where_condition
ORDER BY a.appointment_date DESC, a.appointment_time DESC
    
```

Fig. 41 Staff generate report code

The generate report page in Figure 40 allows staff to view the appointment reports with a filter that is easy for staff to customize. The filter allows staff to select from start date to end date and choose a report type based on preference. The table will show a complete list of appointment details. Staff can print the report or export the CSV for action.

5.2 Testing Result

Table 6 *Testing Result*

Module	Description	Expected Result	Actual Result	Pass/Fail
Login and Registration	To check whether pet owner able to register and login account	Pet owner able to create an account and login	Pet owner successfully created account and login	Pass
	To check whether staff and veterinarian able to login to different system account based on role	Staff and veterinarian able to login to assigned role account using same login page	Staff and veterinarian successfully login to assigned role account using same login page	Pass
	To check whether admin able to create staff and veterinarian accounts	Admin able to create staff and veterinarian account	Admin successfully create staff and veterinarian account	Pass
Appointment	To check whether pet owner can book appointments for each cat	Pet owner able to book appointments for each cat based on availability and fill in appointment details	Pet owner successfully books appointments for each cat based on availability and fill in appointment details	Pass
	To check whether staff able to receive appointments request and assign to selected veterinarian	Staff able to receive appointments request and assign to selected veterinarian	Staff successfully receive appointments request and right veterinarian receive assigned appointment	Pass
	To check whether veterinarian able to accept appointment or decline	Veterinarian able to accept appointment or decline	Veterinarian successfully accept appointment or decline	Pass
Notification	To check whether pet owner receive notification on approve or decline appointment	Pet owner able to receive email notification whether the appointment been approved or decline	Pet owner successfully receives email notification approve or decline with brief details such as assign doctor or decline reason	Pass
	To check whether staff and veterinarian receive notification for a new request	Staff and veterinarian able to receive email notification for a new request	Staff and veterinarian successfully receive email notification for a new request	Pass
Health Record	To check whether veterinarian can key in form medical record base on appointment type and able to view by pet owner and staff	Veterinarian able to key in medical record based on appointment type and save the record to be able be viewed by pet owner and staff	Veterinarian successfully save medical record based on appointment type and pet owner and staff able to view	Pass
Reporting	To check whether staff able to generate reports with correct data	Staff able to generate report relate with appointment and data generated correctly	Appointment report generate correctly	Pass

5.3 User Acceptance Testing

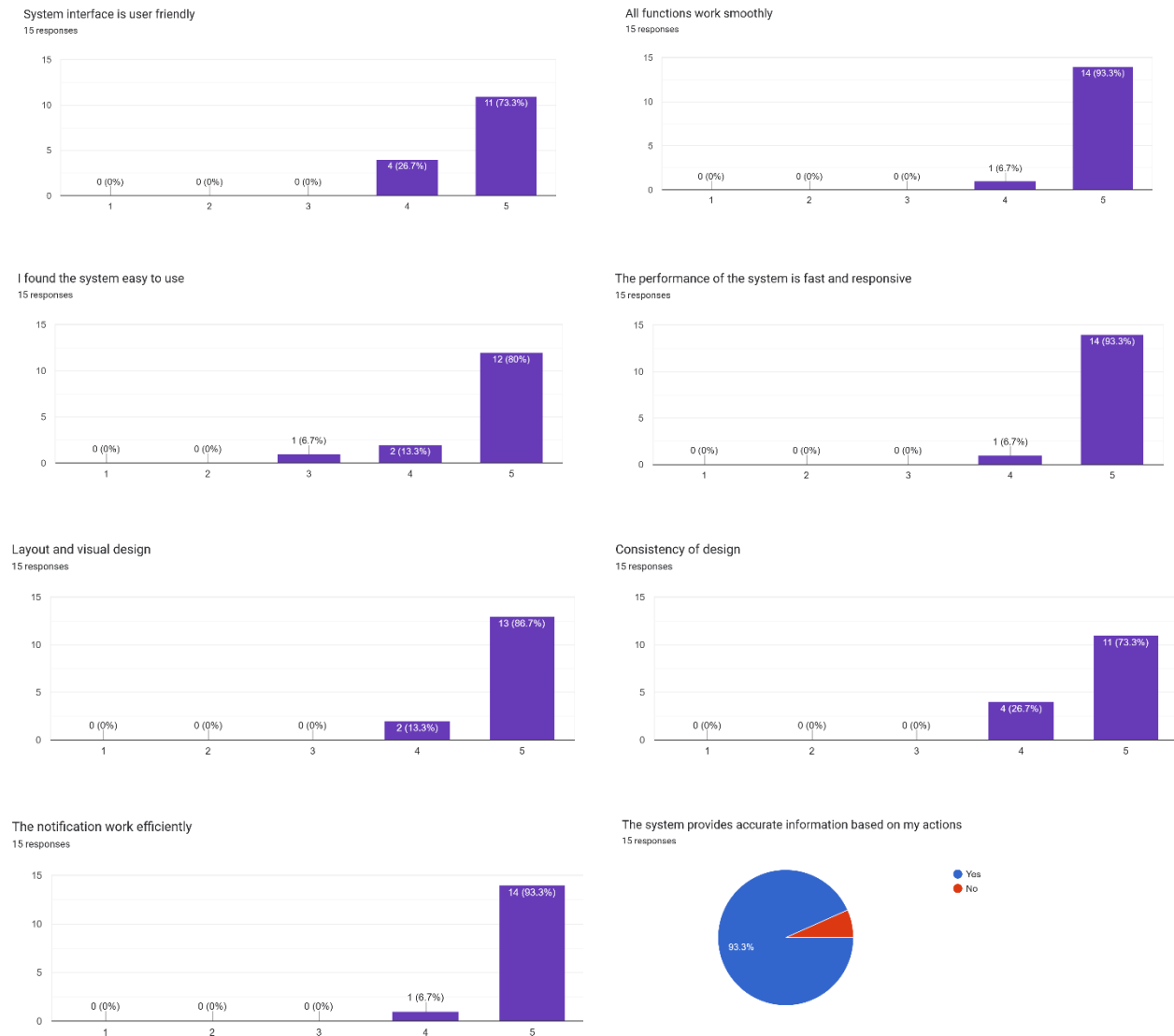


Fig. 42 user Acceptance Testing for GatoPointment: cat Health Appointment System

Figure 42 presents the results of user testing of the GatoPointment System. It receives positive results across all evaluated areas. A total of 15 participants responded to the evaluation and 8 evaluation answer forms were given to them. For the system user interface, 73.3% of respondents rated 5, and 26.7% rated 4 for user friendly. In terms of function work smoothly, 93.3% rated 5 while only 6.7% rated 4. 80% found the system easy to use while only 6.7% rated 3. The design consistency was rated 5 by 73.3% while the layout and visual design was rated 5 by 86.7% of respondents. Most 93.3% of respondents found that the system's performance is fast and responsive and the same goes for notifications that work efficiently. Overall, the system provides accurate information based on user action.

6. Conclusion

This project aims to achieve four main goals for improving both veterinary and pet owners to improve pet health management, veterinary efficiency, streamline appointment scheduling and improve time and resource efficiency. The project allows veterinary clinic staff to dedicate more time and resources to patient care, ultimately leading to better pet health outcomes. By the end of the project, the project should enhance appointment management by giving accurate time allocations for different appointment types and improving clinic scheduling efficiency. Improve the process of managing cat health appointments, allowing a smoother experience for pet owners and clinic staff. Future work will improve the user experience by adding appointments reminder notification, staff able to filter schedule view for selected veterinarian.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

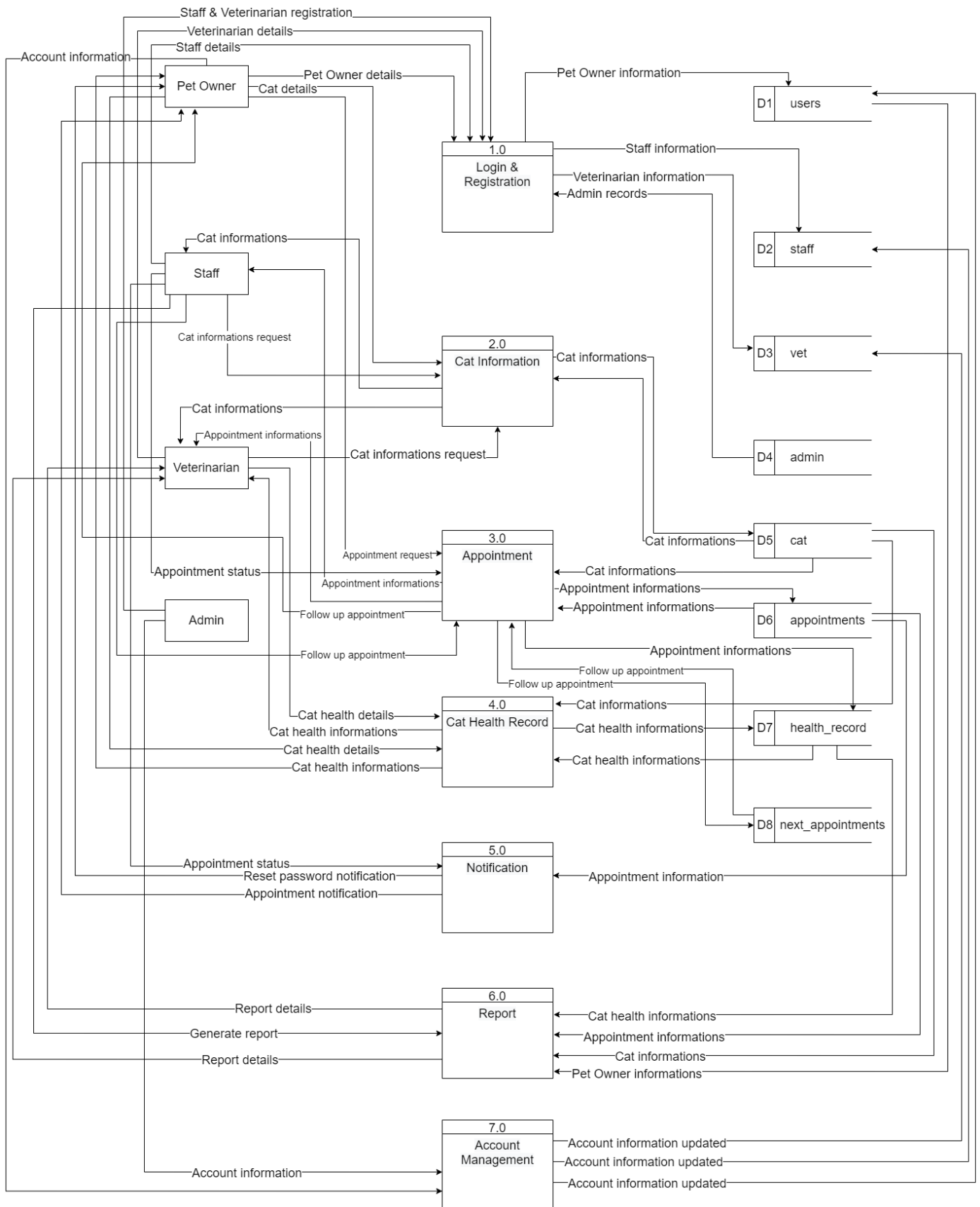
Author Contribution

The authors confirm contribution to the paper as follows: **study conception and design:** Wan Ahmad Waafi Yasin Omar, Rozita Abdul Jalil; **data collection:** Wan Ahmad Waafi Yasin Omar, Rozita Abdul Jalil; **analysis and interpretation of results:** Wan Ahmad Waafi Yasin Omar, Rozita Abdul Jalil; **draft manuscript preparation:** Wan Ahmad Waafi Yasin Omar, Rozita Abdul Jalil. All authors reviewed the results and approved the final version of the manuscript.

References

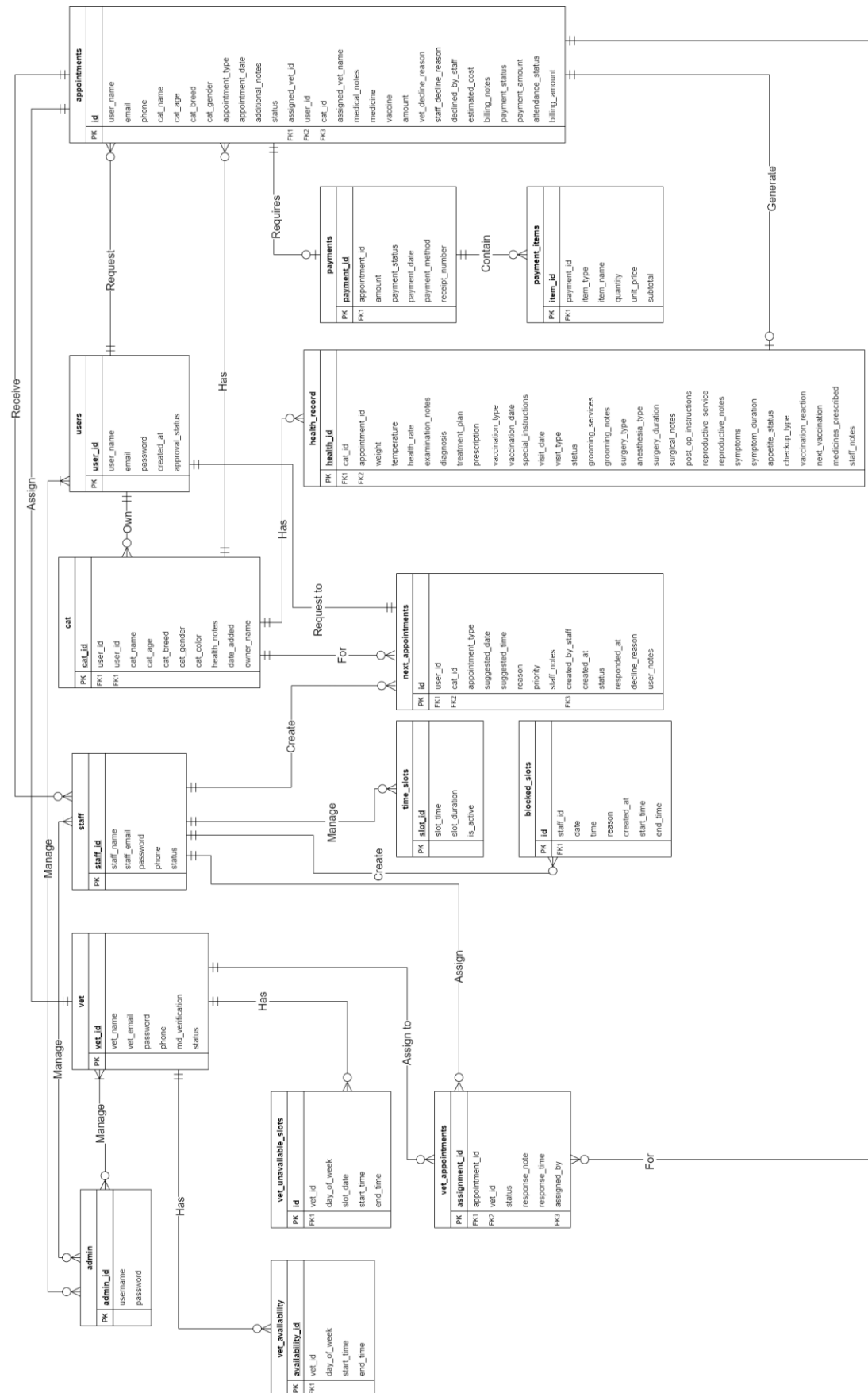
- [1] A. S. Azhari, "Veterinary Clinic Management System," 2020. Available: <https://ir.unimas.my/id/eprint/33985/1/Ainin%20Sofea%20Binti%20Azhari%20-%202024%20pgs.pdf>
- [2] R. Arifah Fasha and M. Muhammad Harith, "An Online Scheduling Platform for Veterinary Appointments," Jurnal Intelek, vol. 18, no. 2, Aug. 2023, doi: <https://doi.org/10.24191/ji.v18i2.22092>. Nicholl, K. (2020, May). A royal spark. *Vanity Fair*, 62(5), 56–65, 100.
- [3] L. Corah, A. Lambert, K. Cobb, and L. Mossop, "Appointment scheduling and cost in first opinion small animal practice," Heliyon, vol. 5, no. 10, p. e02567, Oct. 2019, doi: <https://doi.org/10.1016/j.heliyon.2019.e02567>.
- [4] M. Z. Abidin and M. M. Yusof, "Veterinary Clinic Management Application System," Applied Information Technology And Computer Science, vol. 3, no. 1, pp. 632–648, Jun. 2022, Available: <https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/2344/1903>
- [5] "Online doctor appointment system - Entity-relationship diagram example | Gleek," Gleek.io, 2023. https://www.gleek.io/templates/online-doctor-appointment#google_vignette (accessed Dec. 31, 2024).
- [6] "Book Vet Appointment Online in Veterinary Clinics in Malaysia | Qmed Asia," qmed.asia. <https://qmed.asia/find-care/veterinary>
- [7] "Veterinary Appointment Scheduling Software," Vetstoria. <https://www.vetstoria.com/>
- [8] "Cloud Management System for Vet Clinics & Grooming Centres," kumoVet. <https://kumovet.com/>
- [9] "Kreloses - All-in-one Cloud-based Clinic Management System," Kreloses.com, 2024. <https://www.kreloses.com/> (accessed Dec. 31, 2024).
- [10] K. Keenmon, "Veterinary Record Keeping: Medical Templates, Examples & Laws," Otto, Nov. 16, 2022. <https://otto.vet/veterinary-record-keeping/>

Appendix A: Data Flow Diagram Level 0



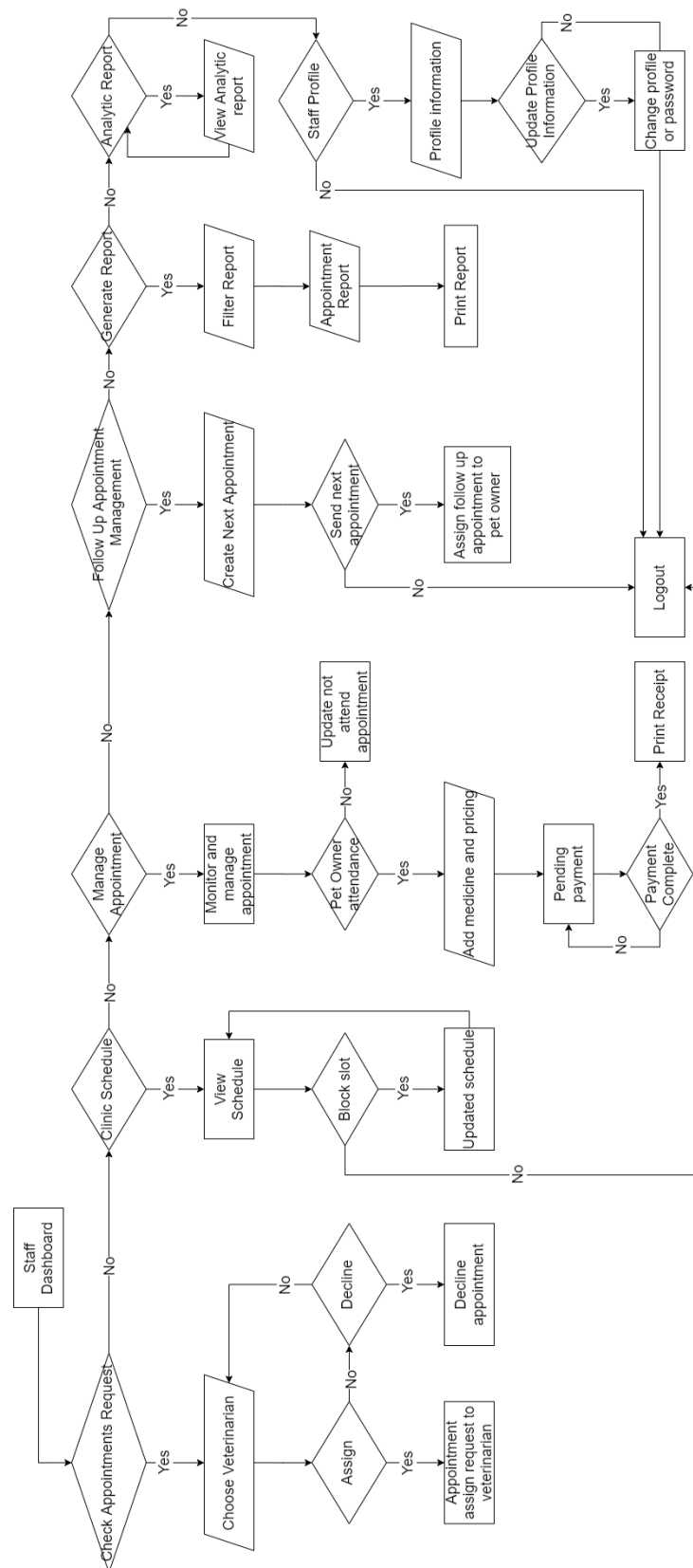
Data Flow Diagram Level 0

Appendix B: Entity Relation Diagram



Entity Relation Diagram

Appendix C: System Flowchart Staff



System Flowchart Staff