

# Dr Ain Veterinary Patient Record Information System (DVAS): An Implementation of Role-Based Access Control and Zero Trust Approach

Nur Syafiqah Abdul Salim<sup>1</sup>, Nurul Hidayah Ab Rahman<sup>1\*</sup>

<sup>1</sup> *Fakulti Sains Komputer dan Teknologi Maklumat,*

*Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

\*Corresponding Author: [hidayahar@uthm.edu.my](mailto:hidayahar@uthm.edu.my)

DOI: <https://doi.org/10.30880/aitcs.2025.06.01.025>

## Article Info

Received: 24 July 2024

Accepted: 19 June 2025

Available online: 30 June 2025

## Keywords

Veterinary sytem, Booking system, Role-Based Access Control, Zero Trust, Appointment Booking System.

## Abstract

Veterinary Information Management System (VIMS) are becoming increasingly essential in modern veterinary practices, streamlining the management of patient's records, scheduling and treatment histories. However, these systems are susceptible to cyberattacks, such as data breaches and unauthorized access attempts. In the event of a security incident, it is crucial to conduct thorough investigations to safeguard sensitive pet health data, the pet owner data and ensure the integrity of the veterinary practice. Therefore, this study proposed a Patient Record Information System for Dr Ain Veterinary with Role-Based Access Control and Zero Trust (DVAS), a VIMS developed with role-based access control and zero trust. The prototyping model was adopted to develop the DVAS system which includes five user modules such as doctors, pharmacists, helpdesk staff, admin and pet's owner. Functional testing results confirm DVAS's ability to manage the 5 user modules, the database connected to the system, logs functionality, errors in invalid inputs, RBAC and zero trust function successfully. The user acceptance test shows positive feedback and high satisfactions, confirming the effectiveness of DVAS's features and performance. To sum up, DVAS demonstrates promising features that protect against data breaches and unauthorized access attempts.

## 1. Introduction

This project focuses on Dr Ain Veterinary, located at Taman Desa Tebrau. Currently, the clinic uses KRELOSES system, that operates with a simple username and password authentication mechanism. This setup lack of necessary for comprehensive security and data privacy in healthcare environment. Patient data, which often includes highly sensitive medical records, is essentially stored in an open-access format. In this system, all users, regardless of their role or responsibilities within the clinic, are provided with the same level of access to patient information.

Dr. Ain Veterinary relies on the KRELOSES system to manage their patient data. However, this system poses several challenges, especially in terms of security, as it only requires login and username, lacking the necessary security measures. This has led to several problems for the clinic:

1. The current system struggles with managing patient records while ensuring the highest level of data security. For example, after staff members log into the system, there are no additional security measures in place.

2. The current system lacks robust role-based access control (RBAC), making patient data vulnerable to unauthorized access. This can end up causing the system to cyberthreats which are data breach, unauthorized access and also SQL injections.
3. The need for confidentiality and privacy in healthcare regardless of the security limitations.

Therefore, this study proposed A Patient Record Information System for Dr Ain Veterinary (DVAS) with Role-Based Access Control and Zero Trust. Addressing the identified problem, the proposed system introduces Role-Based Access Control (RBAC) to effectively segregate roles and restrict access to patient data, thereby enhancing the integrity and confidentiality of sensitive patient information. There are three objectives to this proposed system which are:

1. To design a Patient Record Information System for Dr Ain Veterinary with Role Based Access Control and Zero Trust (DVAS).
2. To develop a Patient Record Information System for Dr Ain Veterinary with Role Based Access Control and Zero Trust (DVAS).
3. To test the proposed system with functionality, users and security testing.

The proposed system aims to ensure the utmost security, confidentiality, and trustworthiness of patient's and pet owner's sensitive data with user trust for the clinic staff's teams. By strengthening the access control and adding one time password this will ensure the protection of patient data even in the case of potential breaches. In conclusion, DVAS aims to rectify the existing vulnerabilities and provide a secure, reliable, and efficient system for managing patient data. By addressing the current security gaps and enhancing data protection measures, the DVAS system will significantly improve the clinic's ability to safeguard patient information, thereby fostering a more secure and trustworthy healthcare environment.

## 2. Literature Review

This section explains the information management system, Role-Based Access Control (RBAC), Zero trust and also comparison of 3 existing system with proposed system.

### 2.1 Information Management System

The Information Management System (IMS) serves as a crucial tool for organizational management, facilitating the transformation of data into useful information presented in the right manner. Effective decision-making within IMS involves five key steps: collecting, processing, storing, retrieving, and disseminating relevant information, all aimed at the efficient management of an organization[1].

These systems not only act as centralized data repositories but also function as facilitators, fostering collaboration among organizational members. Furthermore, IMS plays a pivotal role in streamlining processes throughout an organization by automating routine tasks, data entry, and document handling, thereby reducing time and effort expended in daily operations.

Dr. Ain Veterinary uses KRELOSES Vet Software as their IMS. This system surely needs improvement as the current system only provides a user-friendly interface to the staff, the pet owner cannot book an appointment where they only use a first come first serve method. This method is a waste of time as the owner needs to wait for a long time as pets really need quality care in handling.

This Veterinary also needs a robust IMS to ensure the clinic has streamlined workflows and automated routine tasks. This is because automation takes time, increases production, and improves quality[2]. This will ensure when booking features are added it will simplify the staff workflow and save the quality time of the patient's owner.

### 2.2 Role- Based Access Control

Role-Based Access Control (RBAC) is a method that manages the access of subjects based on their role within the system [4]. In simple terms, a limited number of roles can represent numerous users. This system employs five roles, encompassing doctors, helpdesk staff, pharmacists, pet owners and the system admin. Each role is assigned a distinct level of access to patient information, ensuring the highest level of data privacy.

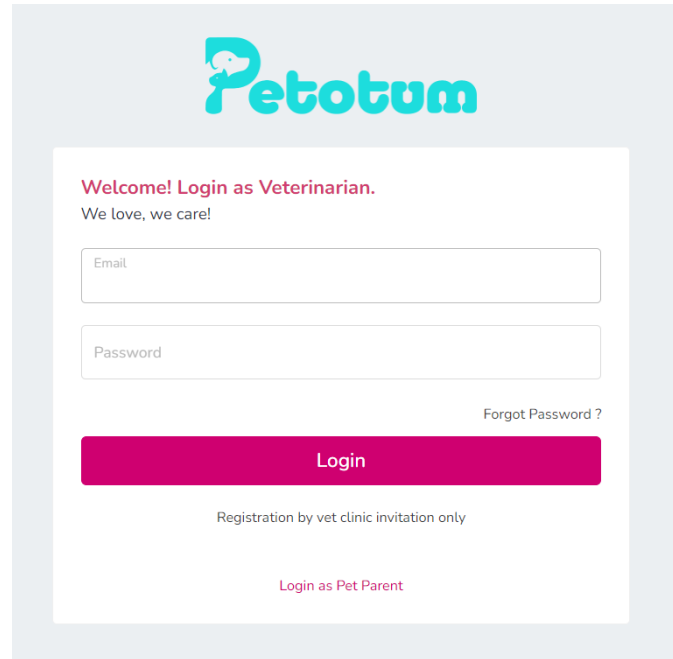
### 2.3 Zero Trust

Zero trust is a security model that is based on the principle of not trusting any user or system by default even if the person is inside of the corporate network [3]. This feature ensures the highest level of security for patient data owners through continuous authentication.

In this system, the role subjected to this authentication process is the doctors, as they require full access to patient data. The access to sensitive data must undergo continuous authentication, involving the input of an eight-digit password sent through email during the login session. Previously, the clinic lacked security features, allowing all staff to access data at the same level.

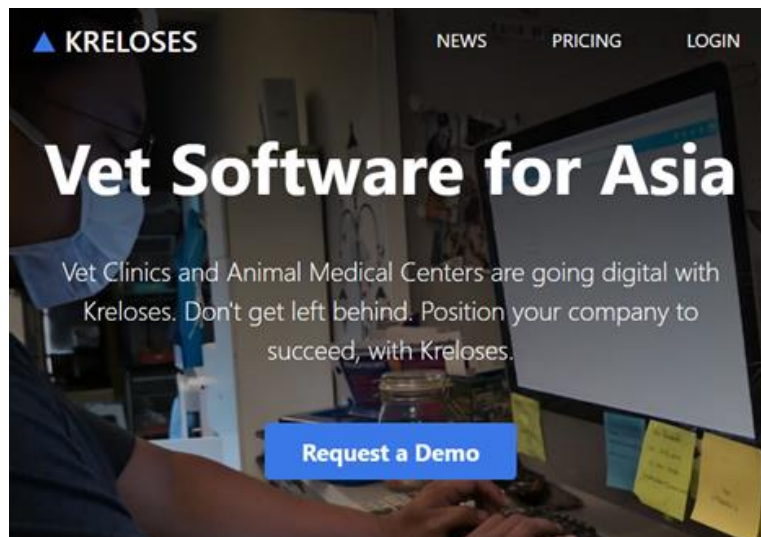
## 2.4 Comparison with existing system

The study of the related system involves analysing several existing systems related to the topic. The study's purpose is to ensure the understanding of how the existing system works and what is the downside and which part need improvements. The comparison between the three existing system will be PETOTUM Vet (PV), KERLOSES Vet Software (KVS) and also Hippo Manager Veterinary Software (HMVS).



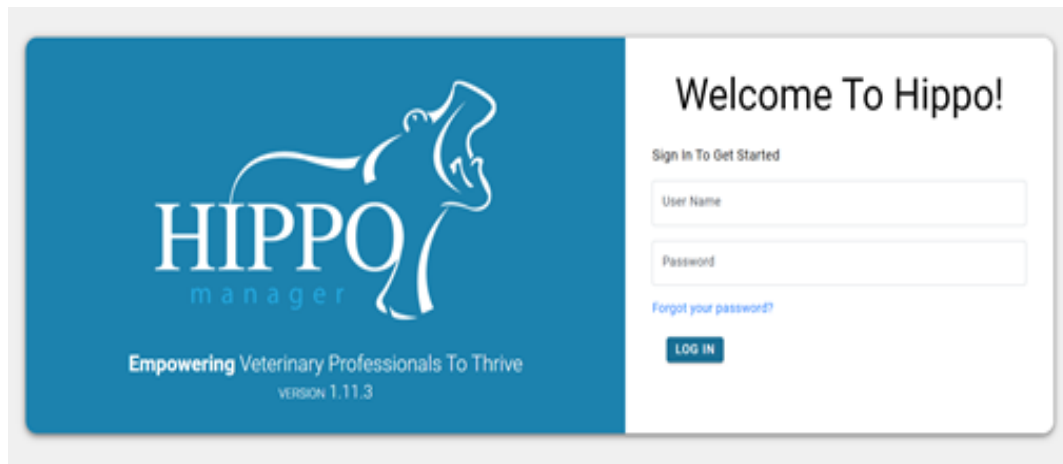
**Fig. 1** PETOTUM Vet interface

Fig. 1 shows the PV which is a web-based application with a user-friendly interface and clear labelling ensuring ease of use. The system is designed to be customizable allowing clinic to tailor it to their specific needs, but the system does not apply any knowledge-based system that lacks pet owner-accessible portal.



**Fig. 2** KERLOSES Vet Software interface

Fig. 2 shows KVS which is cloud-base application which allows customers to customize the platform according to their needs. This software also only has features for clinic staff only. This clinic allows the staff to customize the platform according to their needs but the system do not apply any interface for the pet owner-accessible features which is to book an appointment.



**Fig. 3** Hippo Manager Veterinary Software interface

Fig. 3 shows HMVS which is a cloud-based software that have a simple and user-friendly interface. This feature allow user to navigate easily but this system has lack of reporting features that makes it difficult to the clinic to do any monthly reports.

Table 1 shows the comparison between the three existing system and proposed system. PV indicates as PETOTUM Vet, KV as KRELOSES Vet Software and HMVS as Hippo Manager Veterinary Software with the proposed system which are DVAS. PV and DVAS are web-based applications, while KV and HMVS are cloud-based. All systems support login functionality. However, only DVAS offers password complexity, role-based access control, and zero trust security, features not present in the other systems. For appointment scheduling, PV supports this feature, but KV and HMVS do not, whereas DVAS does. Additionally, PV and DVAS can summarize appointments, a capability absence in KV and HMVS. This comparison highlights that DVAS, the proposed system, offers a more comprehensive set of features, particularly in terms of security and appointment management.

**Table 1** Comparison between existing and proposed system.

System/Features	PV	KV	HMVS	DVAS
Module	Web-based application	Cloud- based application	Cloud- based software	Web-based software
Login	Yes	Yes	Yes	Yes
Password Complexity	No	No	No	Yes
Appointment Scheduling	Yes	No	No	Yes
Role-Based Access Control	No	No	No	Yes
Zero Trust	No	No	No	Yes
Summarize appointment	Yes	No	No	Yes

### 3. Methodology

The DVAS system was developed using software development technique that is a prototyping model. The prototyping model collects information based on user requirements and system development focuses on the user. In this study, prototyping model applied comprises five phases which are planning, analysis, design, implementation and testing phase. The expected outcome of each phase is presented in Table 2.

**Table 2** Phases in this methodology

Phase	Activity	Outcome
Planning	<ul style="list-style-type: none"> <li>Define the problem statement, objectives and scope of the project.</li> <li>Proposed the project title.</li> <li>Study the existing system and analyst the problem and solutions.</li> <li>Create the project planning and schedule.</li> <li>Search more references regarding the existing system.</li> </ul>	<ul style="list-style-type: none"> <li>Project proposal.</li> <li>Gantt Chart.</li> <li>Literature review.</li> </ul>
Analysis	<ul style="list-style-type: none"> <li>Identify the system requirements.</li> <li>Design UML diagrams and STRIDE threat modelling.</li> <li>Research of the proposed system.</li> </ul>	<ul style="list-style-type: none"> <li>Use-case diagram.</li> <li>Class diagram.</li> <li>System architecture.</li> <li>Activity diagram.</li> <li>Abuse case diagram.</li> <li>STRIDE threat modelling.</li> </ul>
Design	<ul style="list-style-type: none"> <li>Design the database.</li> <li>Design user interface.</li> <li>Make a checklist of system testing.</li> </ul>	<ul style="list-style-type: none"> <li>System architecture.</li> <li>Data dictionary.</li> <li>Testing plan.</li> <li>Interface design.</li> </ul>
Implementation	<ul style="list-style-type: none"> <li>Develop and implement a prototype.</li> <li>End users evaluate the system.</li> </ul>	<ul style="list-style-type: none"> <li>System prototype developed.</li> <li>Gain user feedback.</li> </ul>
Testing	<ul style="list-style-type: none"> <li>Conduct the system testing plan.</li> </ul>	<ul style="list-style-type: none"> <li>Status of functionality, security and user testing is updated.</li> <li>Finalize the system.</li> </ul>

The project methodology involves several key phases, starting with the planning phase, which involves activities resulting in a project proposal, Gantt chart, and literature review. In the analysis phase, every necessary diagram is designed to aid in the project's analysis. The design phase follows, focusing on designing the database, system interface, and testing plan. During the implementation phase, a prototype is developed and evaluated by end users, yielding a system prototype and user feedback. Finally, the testing phase involves rigorous testing to ensure the system's functionality, usability, and security, leading to updated results and the finalization of system deployment.

### 3.1 System Architecture

This system architecture will describe the components, modules and interface of DVAS. Fig. 4 shows the system architecture for this proposed system. DVAS includes 5 users: admin, doctors, pharmacists, helpdesk staff, and pet's owner. When the user logs into the system, the system will send a request to the database to verify user's identity before access allowed. After getting access to the system, users can request to manage their modules from the database.

Admin can request to manage staff, pet, vaccination, appointment and payment data. The admin needs to insert an OTP number to ensure the security of the system. Doctors can request to manage, view and print the patient's medical and medicine record but need to insert OTP to verify themselves. Next, pharmacists have the same access as doctors to ensure the patients get the medicine they need. Helpdesk staff can request to manage the patient's booking information and view the booking information. Lastly, pet's owners can book their appointment for their pets

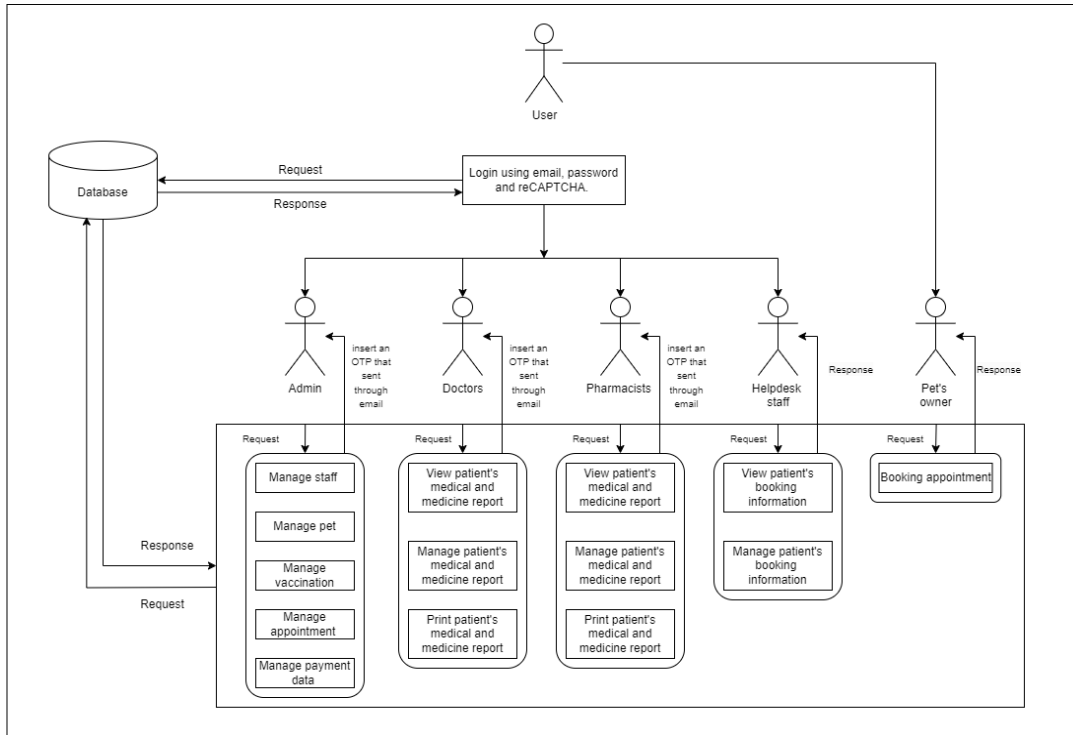


Fig. 4 System architecture of DVAS

### 3.2 Abuse Case Diagram

The proposed abuse case diagram is to identify a possible type of attack that can be performed by the threat actor. Fig. 5 shows three possible threats who are identified as malicious admin, malicious doctors and script kiddies. Malicious admin could potentially misuse the staff information, delete any personal information and edit the monthly report of the clinic. The next potential attack by malicious doctors is editing patients medical record, delete patient's information and steal pet owner personal information. The last threat actor in this abuse case diagram is the script kiddie. The possible attack that might be done by the script kiddie is execute a brute force attack and delete patient's information.

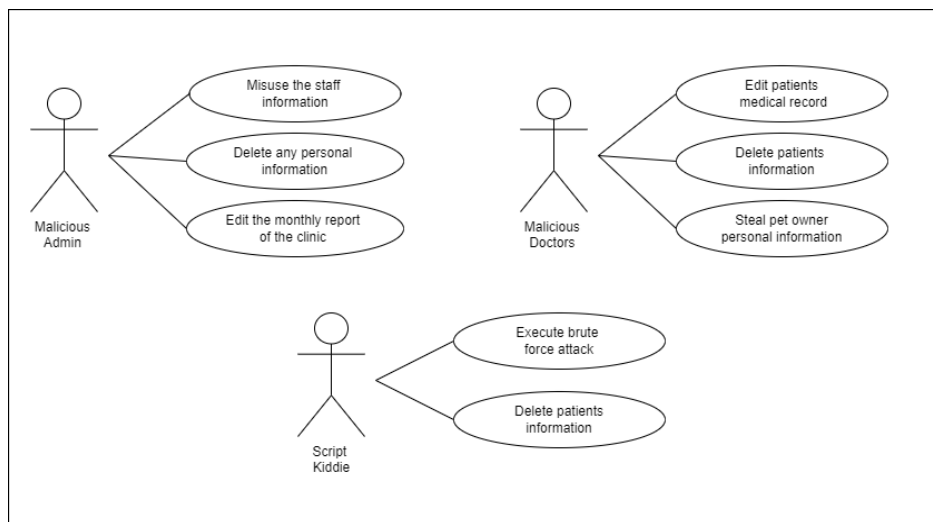


Fig. 5 Abuse Case diagram of the project

### 3.3 STRIDE

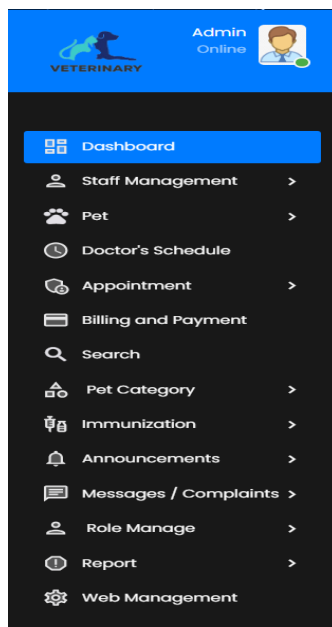
The STRIDE model is used to analyze and determine the potential threats to a system. Table 3 shows the potential threats posed by an attacker for Dr. Ain Veterinary System.

**Table 3** *The potential threat*

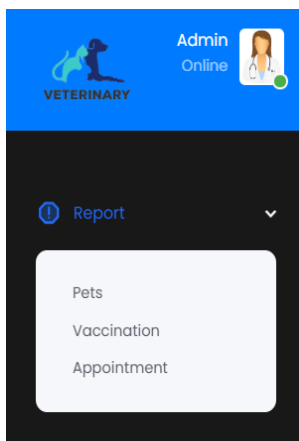
Threat	Potential Attack	Description
Spoofing	Dictionary Attack	An attacker can attempt to guess a user’s password by enter a common or possible password.
Tampering	SQL Injection	SQL injection enables attackers to manipulate system queries, granting unauthorized access to the system, allowing data alteration, table deletion, modification of user information, data destruction, and potential acquisition of database administrator privileges.
Information Disclosure	Identity theft	An attacker gains unauthorized access to the system by stealing the user’s login credentials and commits crime.
Elevation of privilege	Privilege elevation attack	An attacker bypasses the authorization and access as admin to increase privilege results in getting access to change doctors, helpdesk staff or pharmacist’s information.

#### 4. Results and Discussion

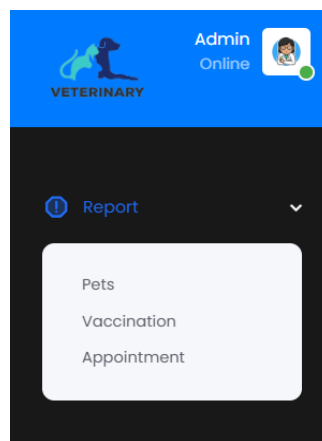
This section explains the analysis and design of this DVAS includes describing the requirements needed, system architecture, abuse case diagram and STRIDE. For the analysis and design, this DVAS implies RBAC and Zero Trust in the system, RBAC implies in management part which makes all the staff have different level of access to ensure the patients data integrity.



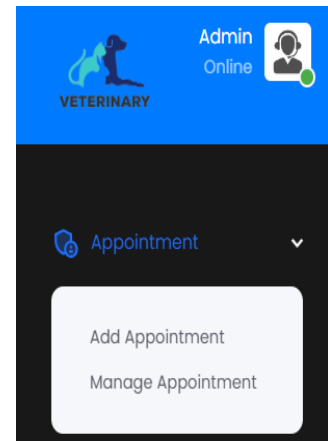
**Fig. 6** Admin interface



**Fig. 7** Doctors



**Fig. 8** Pharmacists



**Fig. 9** Helpdesk staff

*interface**interface**interface*

Fig. 6 shows the admin can manage the whole system includes the staff, pet, appointment, billing receipt and data, pet category, vaccination (immunization), monthly report of the clinic. Fig. 7 and Fig. 8 shows doctors and Pharmacists have the same level of access, this is because doctor who are treating them but pharmacists also have a huge role to ensure the medicine given is accurate. Lastly, Fig. 9 shows the helpdesk staff can manage the booking information and add new bookings for walk-in session patients. As for pet owners they can book their pet appointment by searching for the slot available from the websites.

```

1 <div class="app-sidebar">
2 <div class="logo">
3 <a href="#" class="logo-icon"><span class="logo-text">Appointment</span></a>
4 <div class="sidebar-user-switcher user-activity-online">
5 <a href="#">
6 
7 <span class="activity-indicator"></span>
8 <span class="user-info-text">User<br><span class="user-state-info">Online</span></span>
9 </a>
10 </div>
11 </div>
12 <div class="app-menu">
13 <ul class="accordion-menu">
14 <li class="sidebar-title">
15 Apps
16 </li>
17 <li class="active-page">
18 <a href="dashboard.php" class="active"><i class="material-icons-two-tone">dashboard</i>Dashboard</a>
19 </li>
20 <li>
21 <a href="page_appointment"><i class="material-icons-two-tone">category</i>Category<i class="material-icons has-sub-menu">keyboard_arrow_right</i></a>
22 <ul class="sub-menu">
23 <li>
24 <a href="add-category.php">Add Category</a>
25 </li>
26 <li>
27 <a href="manage-category.php">Manage Category</a>
28 </li>
29 </ul>
30 </li>
31 <li>
32 <a href="#"><i class="material-icons-two-tone">pets</i>Pet<i class="material-icons has-sub-menu">keyboard_arrow_right</i></a>
33 <ul class="sub-menu">
34 <li>
35 <a href="add-pets.php">Add Pets</a>
36 </li>
37 <li>
38 <a href="manage-pets.php">Manage Pets</a>
39 </li>
40 </ul>
41 </li>
42 <li>
43 <a href="#"><i class="material-icons-two-tone">vaccines</i>Immunization<i class="material-icons has-sub-menu">keyboard_arrow_right</i></a>
44 <ul class="sub-menu">
45 <li>
46 <a href="add-immune.php">Add Immunization</a>
47 </li>
48 <li>
49 <a href="manage-immune.php">Manage Immunization</a>
50 </li>
51 </ul>
52 </li>
53 <li>
54 <a href="#"><i class="material-icons-two-tone">notifications</i>Announcements<i class="material-icons has-sub-menu">keyboard_arrow_right</i></a>
55 <ul class="sub-menu">
56 <li>
57 <a href="add-announcement.php">Add Announcements</a>
58 </li>
59 <li>
60 <a href="manage-announcement.php">Manage Announcements</a>
61 </li>
62 </ul>
63 </li>
64 <li>
65 <a href="#"><i class="material-icons-two-tone">chat</i>Messages / Complaints<i class="material-icons has-sub-menu">keyboard_arrow_right</i></a>
66 <ul class="sub-menu">
67 <li>
68 <a href="add-complaint.php">Add Complaint</a>
69 </li>
70 <li>
71 <a href="manage-complaint.php">Manage Complaint</a>
72 </li>
73 </ul>
74 </li>
75 <li>
76 <a href="webmanage.php"><i class="material-icons-two-tone">settings</i>Web Management</a>
77 </li>
78 <li>
79 <a href="#"><i class="material-icons-two-tone">report</i>Report<i class="material-icons has-sub-menu">keyboard_arrow_right</i></a>
80 <ul class="sub-menu">
81 <li>
82 <a href="pet-report.php">Pets</a>
83 </li>
84 <li>
85 <a href="vaccine-report.php">Vaccination</a>
86 </li>
87 <li>
88 <a href="antiremedies-report.php">Mass Anti Radies Vaccination</a>
89 </li>
90 </ul>
91 </li>
92 </ul>
93 </div>
94 </div>
95 </div>
96 </div>

```

**Fig. 10** Code for sidebar.

Fig.10 show the code for the sidebar in this system. The HTML structure represents a sidebar for a web application, including a logo, a user section, and a navigation menu. The navigation items include links to the dashboard, category menu, pet menu, immunization menu, announcements menu, messages and complaints menu, web management link, and the report menu. This user-friendly sidebar allows for easy navigation and features visual indicators to represent and highlight the active page. The system employs Role-Based Access Control (RBAC), so each role has different access to certain pages on the sidebar according to their permissions.

As for zero trust, the system applies these features to admin, doctors and also pharmacists as they are the ones who hold the most sensitive data than the helpdesk staff. Every time they need to edit the record, they need to enter the OTP number that has been sent through their email. Fig.10 shows the interface for user to insert the OTP number for accessing sensitive information and Fig.11 and Fig.12 show the code to apply it.

**Fig. 11** The verification input interface

```

$stmt23 = $db->prepare("SELECT * FROM re_admins where id=?");
$stmt23->execute([$SESSION['id']]);
$result23 = $stmt23->fetch();

$stmt = $db->prepare("SELECT * FROM emailsetting");
$stmt->execute();
$result = $stmt->fetchAll();
// print_r($result);
foreach ($result as $row) {

    $smtp_server = $row['smtp_server'];
    $smtp_password = $row['smtp_password'];
    $smtp_enc = $row['smtp_type'];
    $smtp_username = $row['smtp_username'];
    $smtp_port = $row['smtp_port'];
    $email = $result23['email'];
}

```

**Fig. 12** Code for retrieve SMTP settings

This code involves retrieving the Simple Mail Transfer Protocol (SMTP) settings from the database to ensure proper email sending from the server. It includes detailed steps such as preparing an SQL statement to select all records from the 'emailsetting' table, executing the SQL commands to retrieve the email settings, fetching the results to display all returned data, and extracting the SMTP settings, which contain the address, password, username of the SMTP server, encryption type used, port number, and the recipient's email address.

The PHPMailer is configured with SMTP to send an email to the currently logged-in email address. The code creates a new instance of the PHPMailer class and configures it to use SMTP for sending emails. It sets various SMTP server settings such as the SMTP server address, enabling authentication, username, password, the encryption method used, and the port number for the SMTP server. Next, the code sets the email headers, including the email address, which is the same as the SMTP username. The email content is set by indicating the email body in HTML format and setting the subject to include the current date and time. The HTML content of the email body includes the OTP message. Finally, the code attempts to send the email. If successful, the script proceeds to display the OTP entry form; if not, an error message is displayed.

```

$msg1 = "Dear Staff, <br>
<br>
Your OTP is: <strong>" . $otp. " </strong> <br>
";
// print_r($msg1);exit;

$mail = new PHPMailer(true);

$mail->isSMTP();
$mail->Host = $smtp_server;
$mail->SMTPAuth = true;
$mail->Username = $smtp_username;
$mail->Password = $smtp_password;
$mail->SMTPSecure = $smtp_enc;
$mail->Port = $smtp_port;

$mail->setFrom($smtp_username);
$mail->addAddress($email);

$mail->isHTML(true);
$mail->Subject = 'SYAF_VET ' . $dt;
$mail->Body = $msg1;
$mail->AltBody = $msg1;

if ($mail->send())
// email end

```

**Fig. 13** The PHPMailer configuration.

## 4.1 Requirements

This part focuses on the system's functionality and specifies the features and components in the DVAS Functional requirements are the interaction between the system and its users. Table 4 shows the list of the functional requirements and **Table 5** shows the non-functional requirements:

**Table 4** The functional requirements

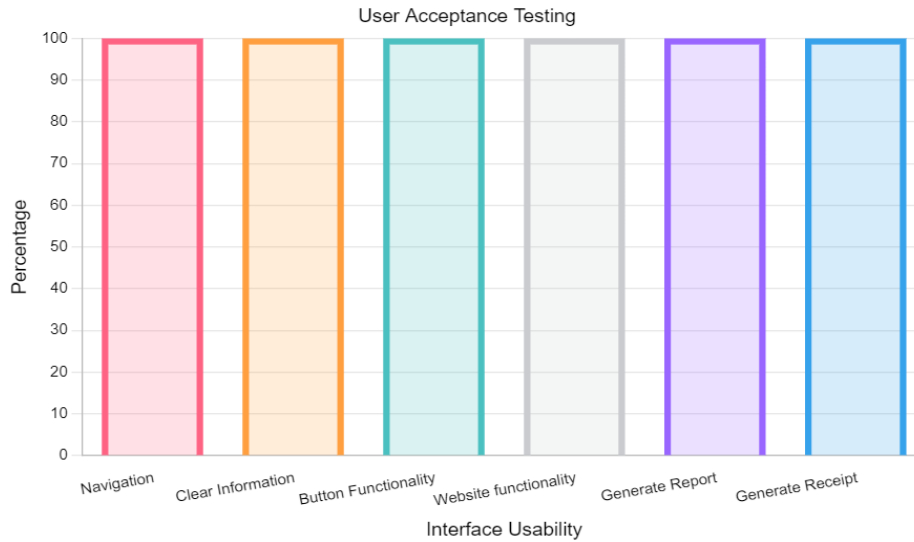
No.	Functional Requirements
1.	The system should be able to provide a login page for the system admin, doctor, helpdesk staff, pharmacists.
2.	The system should allow the admin, doctor, helpdesk staff, pharmacists and patient's guardian to change the password.
3.	System should be able to let doctors view, update and delete the patient's medical detail.
4.	System should be able to let helpdesk staff view a public information of the patients.
5.	System should be able to let pharmacists view and update patient's medicine history.
6.	System should be able to let pet owners book an appointment
7.	The system should be able to confirm the accepted booking.
8.	The system admin will be able to check all the activities on the system.
9.	The system should be able to print monthly receipts of the clinic.

**Table 5** The non-functional requirements

No.	Non-functional Requirements
1.	The system should be able to authenticate and unauthorize user's credential
2.	The system should have password complexity with 8 minimum characters including a combination of alphanumeric, special characters, lowercase and uppercase.
4.	The system should store the booking receipt in the database.
5.	The system should store the user information from registration in database
7.	The system should ask validation before doctors can update or delete patient's medical record.

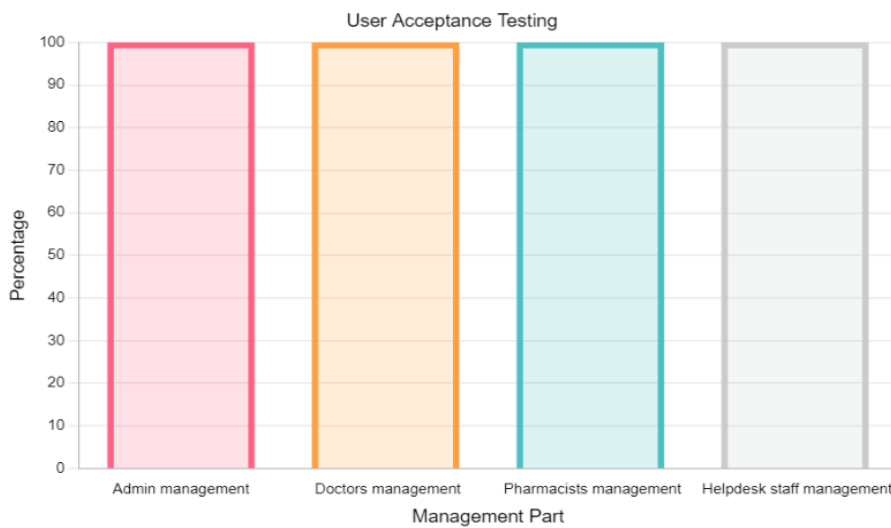
### 4.2 Testing plan

This step is a crucial part to ensure the system functions correctly and meets all the functional and non-functional requirements. This testing part consists of three testing which are functionality, security and user acceptance testing. The results of the testing are crucial to evaluate the system and ensure it meets the objective of DVAS development. These user testing parts consist of interface usability, management part and security features, thirty-one people participate in this testing.



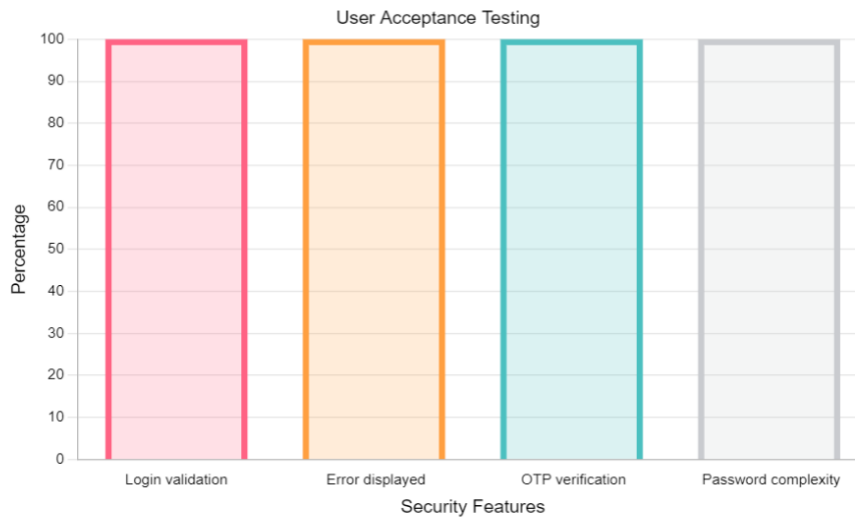
**Fig. 14** *Interface usability*

Fig. 14 shows that all 31 participants agree that the interface usability is good. All the participants confirmed that the system is easy to navigate, the information in the system is clear enough, every button functions properly, the website operates correctly, and the system can generate medical and medicine reports and print billing receipts.



**Fig. 15** *Management par*

Fig. 15 shows the percentages of management part usability as responded by the participants. Users confirmed that admins can manage the whole system, doctors can manage medical records, pharmacists can manage medicine records, and helpdesk staff can manage booking information.



**Fig. 16** Security features

Fig. 16 shows the percentage of security features usability. All the participants confirmed that they can log in to their accounts using the email and password they set. Errors are displayed when the user inputs incorrect information in any part of the requirements. Users also confirmed that admins, doctors, and pharmacists need to insert an OTP number when dealing with sensitive data. They also confirmed that the email is sent to their email account, the same as the login email. Additionally, users confirmed that the system enforces password complexity for their passwords. Table 6 shows the security testing checklist of DVAS that involves testing the system under various scenarios to validate its performance. This testing is to ensure the reliable and user-friendly system.

**Table 6** Security testing of DVAS.

No.	Scenario Test Plan Checklist	Expected Outcome	Actual Output
1.	Data entered must comply with all the data validation criteria.	Only data that comply with the input validation format accepted.	As expected
2.	Users must input OTP numbers to verify themselves before editing the data.	Users that input the valid verification number can access and edit the data. This is the part where zero trust is involved.	As expected
3.	After inputting the OTP verification and turning it to other pages, if want to go back to previous page user need to re-enter the new OTP number.	Involves a strict zero trust features to ensure personal information is safe.	As expected
4.	Admin can delete, add or edit any of the staff or pets and manage the system	Admin can manage all parts of the system but when include sensitive parts there will be an OTP for verification.	As expected

## 5. Conclusion

A Patient Record Information System for Dr Ain Veterinary with Role-Based Access Control and Zero Trust (DVAS) is designed, developed and tested successfully. This section will be the summary part of the system including the advantages, limitations and suggestions for the future work. DVAS has demonstrated the RBAC features throughout all the modules and implemented zero trust features where no one should be trusted even if the system has logged in using the authorized personal email. As a result, the practice can minimize the risk of data breaches and identity theft by unauthorized personnel. However, there are also a few limitations of DVAS that have been identified such as:

1. The system does not have the time limitation for logging out from the system. This is because this part of security features afraid that will make the user more difficult to access to the system as clinic not every time have person in it. So, the system does not have the session timeout but the personal data or sensitive data still secure as user need to insert an OTP number for see the information.

2. The system admin needs to add the staff, pet and vaccination manually to the system individually.  
To improve the functionality and the user experience of the DVAS system, we recommend the implementation of additional features in the future. These suggestions are as follows:
1. Allow the system to have the session timeout at least if one hour inactivity.
  2. Set the system to have own sign in features for any staff.

## 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:** N. S. Abdul Salim, N. H. Ab Rahman; **data collection:** N. S. Abdul Salim, N. H. Ab Rahman; **analysis and interpretation of results:** N. S. Abdul Salim, N. H. Ab Rahman; **draft manuscript preparation:** N. S. Abdul Salim, N. H. Ab Rahman. All authors reviewed the results and approved the final version of the manuscript.

## References

- [1] G. Kiradoo, "Article ID: IJM\_11\_03\_013 Cite this Article: Dr. Giriraj Kiradoo, A Study on Management Information Systems Role and Adoption in Managerial Decision Making," *Int. J. Manag.*, vol. 11, no. 3, pp. 114–121, 2020, [Online]. Available: <http://www.iaeme.com/IJM/index.asp?IType=IJM&VType=11&ITType=3JournalImpactFactor%0Awww.jifactor.comhttp://www.iaeme.com/IJM/issues.asp?IType=IJM&VType=11&ITType=3>
- [2] T. Zayas-Cabán, S. N. Haque, and N. Kemper, "Identifying Opportunities for Workflow Automation in Health Care: Lessons Learned from Other Industries," *Appl. Clin. Inform.*, vol. 12, no. 3, pp. 686–697, 2021, doi: 10.1055/s-0041-1731744.
- [3] U. Mattsson, "Zero Trust Architecture," *Control. Priv. Use Data Assets*, pp. 127–134, 2022, doi: 10.1201/9781003189664-11.
- [4] G. Gurung, R. Shah, and D. P. Jaiswal, "Software Development Life Cycle Models-A Comparative Study," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 3307, pp. 30–37, 2020, doi: 10.32628/cseit206410.
- [5] S. Saeed, N. Z. Jhanjhi, M. Naqvi, and M. Humayun, "Analysis of software development methodologies," *Int. J. Comput. Digit. Syst.*, vol. 8, no. 5, pp. 445–460, 2019, doi: 10.12785/ijcids/080502.
- [6] A. Susanto and Meiryani, "System Development Method with The Prototype Method," *Int. J. Sci. Technol. Res.*, vol. 8, no. 7, pp. 141–144, 2019.
- [7] M. Ali, Yulianti, and E. R. Putra, "Information Systems Stock Inventory in CV Kencana Motor using Laravel Frame," *Tepian*, vol. 2, no. 4, pp. 136–141, 2021, doi: 10.51967/tepian.v2i4.182.
- [8] H. Hustinawati, A. Kurnia Himawan, and L. Latifah, "Performance Analysis Framework Codeigniter and CakePHP in Website Creation," *Int. J. Comput. Appl.*, vol. 94, no. 20, pp. 6–11, 2014, doi: 10.5120/16549-5946.
- [9] M. Clow, "Visual Studio Code," *Angular 5 Proj.*, pp. 57–68, 2018, doi: 10.1007/978-1-4842-3279-8\_5.