

FIND YOUR PAWFECT BUDDY



QUAH JIA YONG

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FIND YOUR PAWFECT BUDDY



This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Database Management) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

FIND YOUR PAWFECT BUDDY

is written by me and is my own effort and that no part has been plagiarized
without citations.

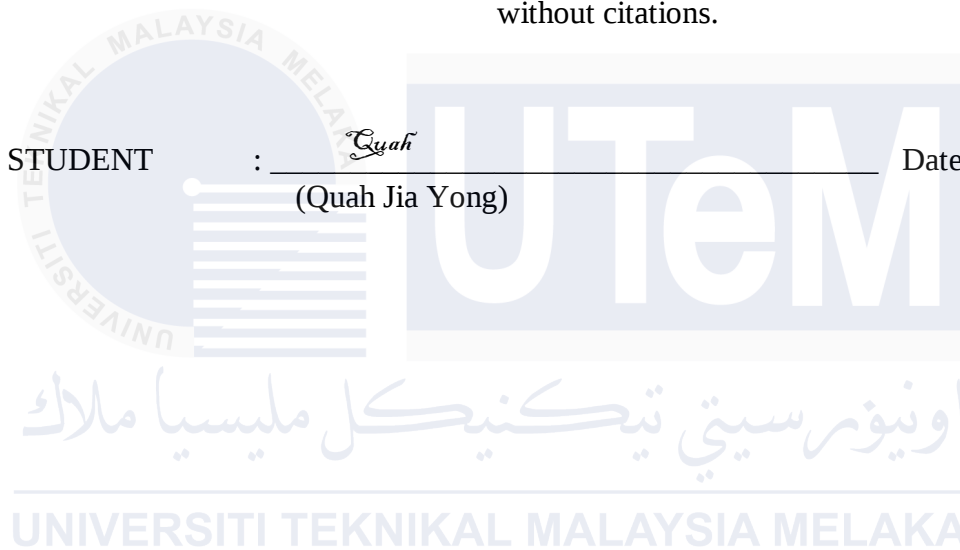
STUDENT

:

Quah

(Quah Jia Yong)

Date : 03/9/2024



I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of Computer Science (Database Management) with Honours.

SUPERVISOR

:

Fathin Nabilla

(Ts. Fathin Nabilla binti Md Leza)

Date : 03/09/2024

DEDICATION

I want to dedicate this project especially to my parents, who I love very much, and my supervisor, who has supported and guided me the entire time. I would especially like to express my sincere gratitude to the following important advisors and contributors.



ACKNOWLEDGEMENTS

I express my gratitude to Ts. Fathin Nabilla binti Md for providing assistance in the successful completion of this project. I am so appreciative of her knowledge, counsel, support, and encouragement. She has consistently and persuasively communicated the spirit of adventure in research and has provided us with insightful direction and encouragement. I also want to express my gratitude to my friend and parents, who have been there for me throughout this project, providing encouragement and support. I would especially like to express my sincere gratitude to the following important advisors and contributors. Finally, but just as importantly, I would like to express my gratitude to my friends, lecturers, and the Faculty of Information and Communications Technology at Universiti Teknikal Malaysia Melaka for all of their help and advice with this report. Without their assistance, this report would have remained a sci-fi fantasy.

ABSTRACT

The issue of stray animals in Malaysia is massive. Millions of dogs and cats wander homeless where there are no homes for them to roam in this system, which we shall henceforth call the "Find Your Pawfect Buddy" project, seeks to revolutionize the way stray animals are reported, rescued and later adopted by pet owners. Some identified challenges that include: uncoordinated report systems; struggles in finding suitable homes for these animals after rescue; and lack of centralized information on the population of strays— these issues will be addressed by a user-friendly platform that eases reporting processes, an adoption system that ensures transparency and data collection aimed at better resource allocation plus planning strategies. The choice of the Waterfall methodology with its structured approach was chosen for a well-defined development process that is stable in nature, yet it caters to a wide range of users including public adopters and shelter staff with different integrated functionalities like animal profiling or adoption management alongside reporting features. The anticipated outcomes include a reduction in stray animals, fewer road accidents caused by strays and improved social conditions in neighborhoods. Finally, "Find Your Pawfect Buddy" aims to protect animal rights and foster a harmonious coexistence between humans and animals through effective stray animal management and adoption solutions.

ABSTRAK

Isu haiwan terbiar di Malaysia adalah besar. Berjuta-juta anjing dan kucing berkeliaran tanpa tempat tinggal di mana tiada rumah untuk mereka berkeliaran dalam sistem ini. Oleh sebab itu, lahirnya projek "Find Your Pawfect Friend", berusaha untuk merevolusikan cara haiwan terbiar dilaporkan, diselamatkan dan kemudian dipelihara oleh pengangkat haiwan. Beberapa cabaran yang dikenal pasti termasuk seperti sistem laporan tidak diselaraskan, bergelut dalam mencari rumah yang sesuai untuk haiwan ini selepas menyelamatkan dan kekurangan maklumat terpusat tentang populasi sesat—isu ini akan ditangani oleh platform mesra pengguna yang memudahkan proses pelaporan, sistem penerimaan yang memastikan ketelusan dan pengumpulan data yang bertujuan untuk peruntukan sumber yang lebih baik serta strategi perancangan. Pilihan metodologi Waterfall dengan pendekatan berstruktur telah dipilih untuk proses pembangunan yang jelas yang bersifat stabil, namun ia memenuhi pelbagai pengguna termasuk pengguna awam dan kakitangan tempat perlindungan dengan fungsi bersepadu yang berbeza seperti latar belakang haiwan atau pengurusan haiwan peliharaan bersama ciri pelaporan. Hasil yang dijangkakan termasuk pengurangan haiwan terbiar, lebih sedikit kemalangan jalan raya yang disebabkan oleh haiwan terbiar yang sesat, dan keadaan sosial yang bertambah baik di kawasan kejiranan. Akhir sekali, "Find Your Pawfect Buddy" bertujuan untuk melindungi hak haiwan dan memupuk kewujudan bersama yang harmoni antara manusia dan haiwan melalui pengurusan haiwan terbiar dan penyelesaian pengambilan yang berkesan.

TABLE OF CONTENTS

DECLARATION	II
DEDICATION	III
ACKNOWLEDGEMENTS.....	IV
ABSTRACT	V
ABSTRAK.....	VI
TABLE OF CONTENTS.....	VII
TABLE OF FIGURES.....	XI
LIST OF TABLES	XIV
LIST OF ABBREVIATIONS.....	XVII
CHAPTER 1: INTRODUCTION.....	1
1.1 Project Background.....	1
1.2 Problem Statement	1
1.3 Objective.....	2
1.4 Scope	2
1.5 Project Significance/Expected Output.....	3
1.6 Conclusion	4
CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY..	5
2.1 Introduction.....	5
2.2 Domain	5
2.3 Existing system	6
2.3.1 Comparison of Existing System.....	6
2.4 Project Methodology	7

2.4.1	Justification	7
2.4.2	Stages	7
2.5	Project Requirements	8
2.5.1	Software Requirement	8
2.5.2	Hardware Requirement	8
2.6	Conclusion	9
CHAPTER 3: ANALYSIS		10
3.1	Introduction.....	10
3.2	Current Scenario Analysis	10
3.2.1	Existing Scenario.....	10
3.2.2	Proposed Application	12
3.3	Requirement Analysis	18
3.3.1	Functional Requirement.....	18
3.3.2	Non-Functional Requirements	19
3.3.3	Software Requirement	19
3.3.4	Hardware Requirement	19
3.4	Project Schedule and Milestones	20
3.5	Conclusion	21
CHAPTER 4: DESIGN.....		22
4.1	Introduction.....	22
4.2	System Architecture Design	22
4.2.1	Scene Sequence Diagram.....	22
4.2.2	High Level Design.....	24
4.2.3	Technology Stack.....	26
4.3	Database Design.....	26

4.3.1	Conceptual Design.....	27
4.3.2	Logical Design	31
4.3.3	Data Model.....	32
4.3.4	Collection Structures	32
4.3.5	Relationships and data linking	34
4.3.6	Key Design Decisions.....	34
4.4	Graphical User Interface (GUI) Design	34
4.4.1	Navigation Design	35
4.4.2	Input Design.....	36
4.4.3	Output Design	43
4.5	Conclusion	65
CHAPTER 5: IMPLEMENTATION.....		66
5.1	Introduction.....	66
5.2	System Development Environment Setup.....	66
5.2.1	Steps of Installation Setup	67
5.3	Database Implementation	70
5.3.1	Create Table Commands.....	70
5.3.2	Insert Data into Collections.....	71
5.3.3	Complex Queries	78
5.3.4	Database Deployment	81
5.4	Conclusion	87
CHAPTER 6: TESTING		88
6.1	Introduction.....	88
6.2	Test Plan	88
6.2.1	Test Environment	90

6.2.2	Test Schedule	91
6.3	Test Strategy	92
6.3.1	Classes of Tests.....	93
6.4	Test Design	94
6.4.1	Test Description	94
6.4.2	Test Data	132
6.4.3	Test Result and Analysis.....	140
6.5	Conclusion	157
CHAPTER 7: CONCLUSION		158
7.1	Introduction.....	158
7.2	Observation Weakness and Strengths	158
7.2.1	Strengths	158
7.2.2	Weaknesses	159
7.3	Propositions of improvement.....	160
7.4	Contribution	160
7.5	Conclusion	161
REFERENCES		162

TABLE OF FIGURES

Figure 1.4.1 System Module	3
Figure 3.2.2.1 Non-Existing Application Flowchart	12
Figure 3.2.2.2 Report Animal Flowchart	13
Figure 3.2.2.3 Manage Saved Animal.....	13
Figure 3.2.2.4 Manage Adoption Process Flowchart	14
Figure 3.2.2.5 Manage Transaction Flowchart	15
Figure 3.2.2.6 Manage Moment Flowchart	15
Figure 3.2.2.7 Manage Treatment Flowchart	16
Figure 3.2.2.8 Account Management Flowchart.....	17
Figure 3.2.2.9 Analysis flowchart	17
Figure 3.3.4.1 Gantt Chart	20
Figure 4.2.1.1 Sequence Diagram.....	23
Figure 4.2.2.1 User Wireframes.....	24
Figure 4.2.2.2 Staff Wireframes	25
Figure 4.3.1.4 DFD Level 2 Report Stray Animal.....	28
Figure 4.3.1.5 DFD Level 2 Animal Profile	28
Figure 4.3.1.6 DFD Level 2 Manage Appointment	29
Figure 4.3.1.7 DFD Level 2 Manage Treatment	29
Figure 4.3.1.8 DFD Level 2 Manage Animal Moment.....	29
Figure 4.3.1.9 DFD Level 2 Manage Fee	30
Figure 4.3.1.10 DFD Level 2 Analysis	30
Figure 4.3.2.1 System ERD	31
Figure 4.4.1.1 Navigation Paths.....	35
Figure 4.4.2.1 User Login.....	36
Figure 4.4.2.2 User Register.....	36
Figure 4.4.2.3 Schedule Appointment	37
Figure 4.4.2.4 Report Stray Animal	37
Figure 4.4.2.5 Appointment Information	38
Figure 4.4.2.6 User Update Details.....	38
Figure 4.4.2.7 Staff Login	39
Figure 4.4.2.8 User Login.....	39
Figure 4.4.2.9 User Register.....	40

Figure 4.4.2.10 Schedule Appointment	40
Figure 4.4.2.11 Report Stray Animal	41
Figure 4.4.2.12 Appointment Information	42
Figure 4.4.2.13 User Update Details	42
Figure 4.4.2.14 Staff Login	43
Figure 4.4.3.1 Update Report Status	43
Figure 4.4.3.2 Report Detail.....	44
Figure 4.4.3.3 Register New Staff	44
Figure 4.4.3.4 Update Staff Detail	45
Figure 4.4.3.5 Animal Profile Update.....	46
Figure 4.4.3.6 Add Animal.....	47
Figure 4.4.3.7 Update Treatment	47
Figure 4.4.3.8 Add treatment.....	48
Figure 4.4.3.9 Update Moment	49
Figure 4.4.3.10 Add Moment	49
Figure 4.4.3.11 Add Transaction	50
Figure 4.4.3.12 Adoption Detail Update.....	50
Figure 4.4.3.13 Browse Animals	51
Figure 4.4.3.14 Animal Details	52
Figure 4.4.3.15 Appointment Information	52
Figure 4.4.3.16 Latest Report	53
Figure 4.4.3.17 Report Detail.....	53
Figure 4.4.3.18 View Animal	54
Figure 4.4.3.19 Animal Profile Detail	54
Figure 4.4.3.20 Medical Record	55
Figure 4.4.3.21 Treatment Detail.....	55
Figure 4.4.3.22 Moment Record	56
Figure 4.4.3.23 Appointment List.....	56
Figure 4.4.3.24 Analysis	57
Figure 4.4.3.25 Analysis Dashboard	57
Figure 4.4.3.26 Pie chart filter by reported stray Animal types.....	58
Figure 4.4.3.27 Pie chart filter by reported stray Animal types.....	58
Figure 4.4.3.28 Pie chart filter by saved animal types	59
Figure 4.4.3.29 Pie chart filter by adopted animal types	59

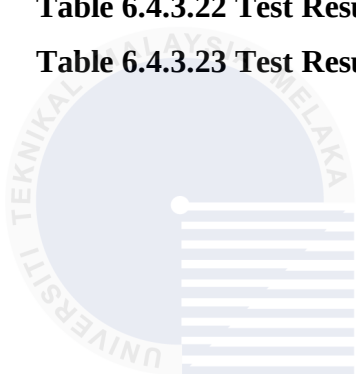
Figure 4.4.3.30 Stray Animal Trends Yearly View Chart	60
Figure 4.4.3.31 Stray Animal Trends Monthly View Selection	60
Figure 4.4.3.32 Adopted Animal Chart	61
Figure 4.4.3.33 Transaction Amount Chart	61
Figure 4.4.3.34 Exported Stray Animal Trends PDF	62
Figure 4.4.3.35 Stray Animal Distribution Map Location Search	62
Figure 4.4.3.36 Stray Animal Distribution for Dog	63
Figure 4.4.3.37 Stray Animal Distribution for Cat	63
Figure 4.4.3.38 Stray Animal Distribution Map Report Status Checked	64
Figure 4.4.3.39 Stray Animal Distribution Map Report Status Unchecked	64
Figure 4.4.3.40 Stray Animal Distribution Map pop up from map points	65
Figure 5.2.1.1 Download MongoDB for windows	67
Figure 5.2.1.2 Startup Screen	67
Figure 5.2.1.3 Service Configuration	68
Figure 5.2.1.4 Choose Setup Type	68
Figure 5.2.1.5 Service Configuration	69
Figure 5.2.1.6 Install MongoDB	69
Figure 5.2.1.7 Installing Process	70
Figure 5.3.3.1 Exclude Feed Paid Query	79
Figure 5.3.3.2 Analysis Complex Queries	80
Figure 5.3.4.1 MongoDB Login	81
Figure 5.3.4.2 Cluster Configuration	82
Figure 5.3.4.3 User Configuration	82
Figure 5.3.4.4 IP access list	83
Figure 5.3.4.5 Cluster Dashboard	83
Figure 5.3.4.6 Cluster Connection	84
Figure 5.3.4.7 Connection String setup	84
Figure 5.3.4.8 MongoDB Compass Connection	85
Figure 5.3.4.9 Create Database	85
Figure 5.3.4.10 Create Collection	86
Figure 5.3.4.11 Insert Document	86

LIST OF TABLES

Table 3.3.4.1 Schedule Table	20
Table 5.3.4.1 User Responsibilities List.....	89
Table 6.2.1.1 Test Environment Hardware List	90
Table 6.2.1.2 Test Environment Software List	90
Table 6.2.2.1 Test Schedule.....	91
Table 6.2.2.1 Type of test and test design techniques for white box and black box testing	93
Table 6.4.1.1 Test Description of User Login	95
Table 6.4.1.2 Test Description of Browse Animals	96
Table 6.4.1.3 Test Description of Animal Details Page	98
Table 6.4.1.4 Test Description of Report Stray Animal Form	100
Table 6.4.1.5 Test Description of Appointment Page.....	102
Table 6.4.1.6 Test Description of Update User Details Page	104
Table 6.4.1.7 Test Description of Staff Login Page.....	105
Table 6.4.1.8 Test Description of Latest Report Page.....	106
Table 6.4.1.9 Test Description of Register New Staff Page.....	108
Table 6.4.1.10 Test Description of Update Staff Details Page	109
Table 6.4.1.11 Test Description of View Animal Page	111
Table 6.4.1.12 Test Description of Animal Profile Update Page	114
Table 6.4.1.13 Test Description of Add Animal Page	115
Table 6.4.1.14 Test Description of Medical Record Page	117
Table 6.4.1.15 Test Description of Add Treatment Page.....	118
Table 6.4.1.16 Test Description of Update Treatment Page	120
Table 6.4.1.17 Test Description of View Moment Page	121
Table 6.4.1.18 Test Description of Add Moment Page.....	123
Table 6.4.1.19 Test Description of Update Moment Page.....	125
Table 6.4.1.20 Test Description of Adoption Page	127
Table 6.4.1.21 Test Description of View Transaction Page	128
Table 6.4.1.22 Test Description of Add Transaction Page.....	129
Table 6.4.1.23 Test Description of Analysis Page.....	130
Table 6.4.2.1 Test Data of User Login Page	132
Table 6.4.2.2 Test Data of Browse Animals Page.....	132

Table 6.4.2.3 Test Data of Animal Details Page	132
Table 6.4.2.4 Test Data of Report Stray Animal Page.....	133
Table 6.4.2.5 Test Data of Appointment Information Page.....	133
Table 6.4.2.6 Test Data of Update User Details Page.....	133
Table 6.4.2.7 Test Data of Staff Login Page	134
Table 6.4.2.8 Test Data of Latest Report Page.....	134
Table 6.4.2.9 Test Data of Register New Staff Page.....	134
Table 6.4.2.10 Test Data of Update Staff Details Page	134
Table 6.4.2.11 Test Data of View Animal Page	135
Table 6.4.2.12 Test Data of Animal Profile Update Page	135
Table 6.4.2.13 Test Data of Add Animal Page	135
Table 6.4.2.14 Test Data of Medical Record Page	135
Table 6.4.2.15 Test Data of Add Treatment Page	136
Table 6.4.2.16 Test Data of Update Treatment Page	136
Table 6.4.2.17 Test Data of View Moment Page	136
Table 6.4.2.18 Test Data of Add Moment Page.....	137
Table 6.4.2.19 Test Data of Update Moment Page.....	137
Table 6.4.2.20 Test Data of Adoption Page	138
Table 6.4.2.21 Test Data of View Transaction Page	138
Table 6.4.2.22 Test Data of Add Transaction Page.....	138
Table 6.4.2.23 Test Data of Analysis Page.....	139
Table 6.4.2.24 Test Data of Analysis Page Chart Filter.....	139
Table 6.4.3.1 Test Result of User Login	140
Table 6.4.3.2 Test Result of Browse Animals Page	140
Table 6.4.3.3 Test Result of Animal Details Page	141
Table 6.4.3.4 Test Results of Report Stray Animal Page.....	142
Table 6.4.3.5 Test Result of Appointment Page	142
Table 6.4.3.6 Test Result of Update User Details	143
Table 6.4.3.7 Test Result of Staff Login	144
Table 6.4.3.8 Test Result of Latest Report Page	144
Table 6.4.3.9 Test Result of Register New Staff Page	145
Table 6.4.3.10 Test Result of Update Staff Details Page	146
Table 6.4.3.11 Test Result of View Animal Page	147
Table 6.4.3.12 Test Result of Animal Profile Update Page.....	148

Table 6.4.3.13 Test Result of Add Animal Page.....	148
Table 6.4.3.14 Test Result of Medical Record Page.....	149
Table 6.4.3.15 Test Result of Add Treatment Page	150
Table 6.4.3.16 Test Result of Update Treatment Page.....	151
Table 6.4.3.17 Test Result of View Moment Page.....	151
Table 6.4.3.18 Test Result of Add Moment Page	152
Table 6.4.3.19 Test Result of Update Moment Page	153
Table 6.4.3.20 Test Result of Adoption Page.....	154
Table 6.4.3.21 Test Result of View Transaction Page.....	155
Table 6.4.3.22 Test Result of Add Transaction Page	155
Table 6.4.3.23 Test Result of Analysis Page	156



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF ABBREVIATIONS

DFD	-	Data Flow Diagram
ERD	-	Entity Relationship Diagram
GUI	-	Graphical User Interface design
SQL	-	Structured Query Language
NOSQL-		Non-structured Query Language
UI	-	User Interface



CHAPTER 1: INTRODUCTION

1.1 Project Background

In 2023, a study from the University of Malaya shows that Malaysia has 6 million stray dogs and 5 million stray cats waiting to be rescued. This number is very disheartening because animals also live. With the urbanization of society, stray animals are neglected and pushed aside. This propose that the trend of stray animals is increasing which the society has bad perception with stray animals while the fact is this stray animal is the victim of the situation. The main reason stray animal happens was because of some irresponsible animal adopter that neglected their animals causing them to become stray animal and loitering around the neighborhood. This can cause harm to the community and the stray animal themselves.

1.2 Problem Statement

1. Inefficient Stray Animal Reporting: Existing methods for reporting stray animals might be fragmented, leading to missed opportunities for rescue and reunification with owners.
2. Difficulty Finding Homes for Strays: Many stray animals struggle to find permanent homes, leading to overcrowding in shelters and euthanasia.
3. Lack of Data on Stray Animal Populations: Without a centralized system for reporting stray animals, it's difficult to track trends in location, species, or numbers, hindering effective resource allocation and targeted rescue efforts.

1.3 Objective

1. To improve seamless and fast accessibility of stray animal reporting by creating a simple and efficient way for people to report stray animals, allowing for faster response times and improved rescue efforts.
2. To increase transparency of adoption process for stray animals from animal profiling to adoption for potential adopters by facilitating a user-friendly platform for connecting potential adopters with adoptable animals, ultimately increasing the number of successful adoptions.
3. To collect and leverage analysis of stray animals reports and adoption records to provide valuable insights into location trends, species most affected, and overall population patterns for better resource allocation and strategic planning.

1.4 Scope

The users are:

- a. Adopter: Individuals seeking to adopt a stray animal.
- b. Public users: People who encounter stray animals and need to report them.
- c. Animal Shelters staff: Partners who can utilize the platform to showcase adoptable animals and manage adoption processes.

Figure 1.4.1 shows the modules for the users and their functions. There is public user, shelter staff, adopter. The public user has stray animal reporting module. The shelter staff has account management module, adoption management module and reporting module. Lastly, adopter has account management module and adopt animal module.

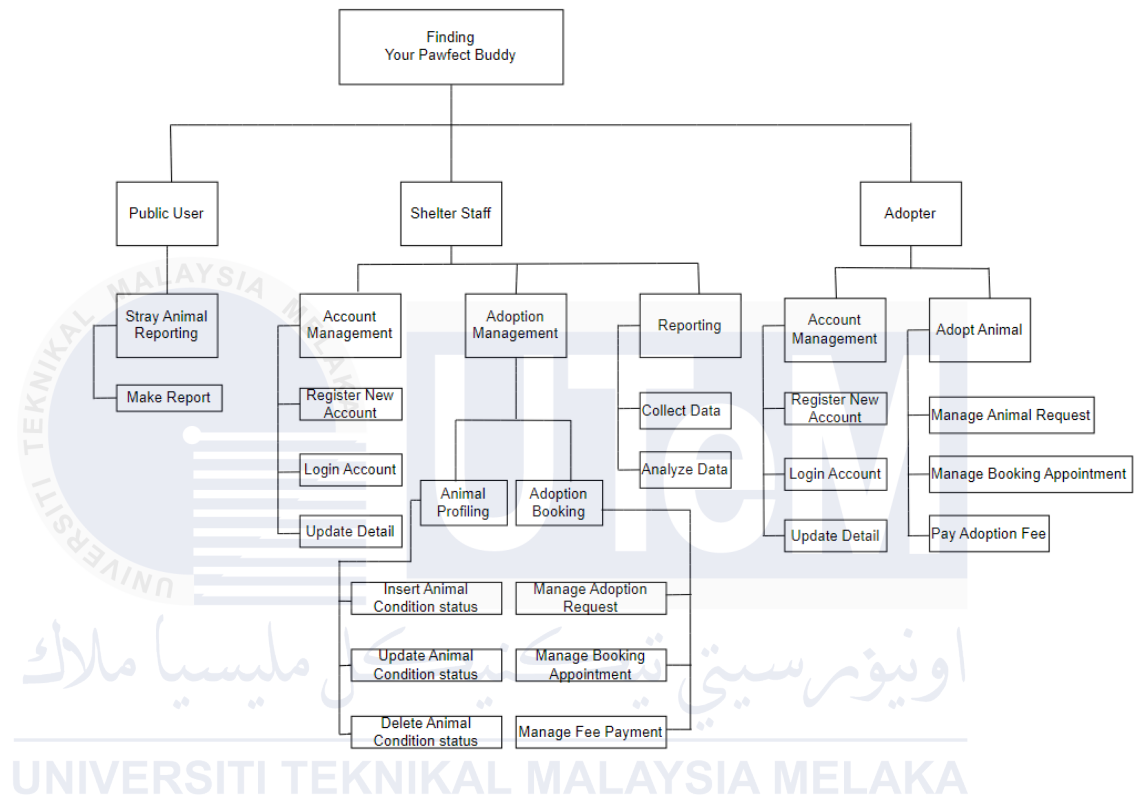


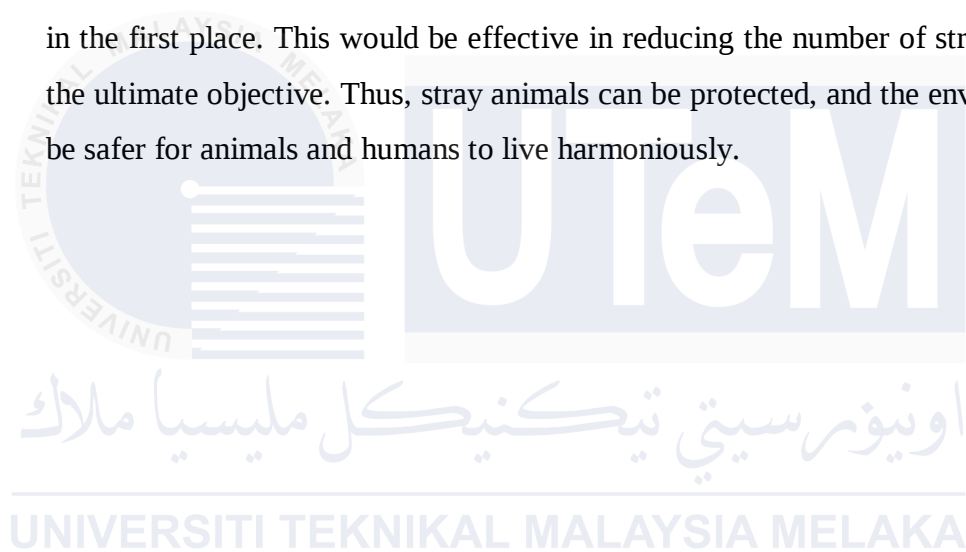
Figure 1.4.1 System Module

1.5 Project Significance/Expected Output

The project's significance is that it helps reduce the number of stray animals. This happens when stray animals can be adopted by a public that loves animals. Also, it can avoid accidents caused on the road by stray animals. Stray animals loitering on the road and unaware of vehicles cause road accidents. Next, it can solve social issues caused by stray animals. It is because some neighborhoods with lower property values and an unsafe environment caused by stray animals can be tackled and reduced with proper measures. Thus, making the area safer for humans and animals. Lastly, stray animals are part of the living beings of the earth; their rights can be protected and saved, which reduces stray animal suffering and neglect as the animal and human relationship is always close together.

1.6 Conclusion

In conclusion, “Find Your Pawfect Buddy” is the perfect stray animal and adoption system that addresses the current issues of stray animals increasing in number and provides a complete solution to provide care to the stray animals and the people in the community. The system is able to consider various angles, including reporting, profiling, and adopting, which is a one-stop solution for stray animal handling. This can help to efficiently manage the stray animal reports and save the stray animal on time. Also, with the analytics tool it proposed, it was able to come up with mitigation steps like raising awareness of taking care of your own pets to prevent stray animals in the first place. This would be effective in reducing the number of stray animals as the ultimate objective. Thus, stray animals can be protected, and the environment can be safer for animals and humans to live harmoniously.



CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

There is serious concern over Malaysia's expanding stray animal population. Effective solutions are required to manage the issue of millions of strays roaming the country and to guarantee the welfare of these animals. The literature on stray animal management and websites for animal shelters is examined in this chapter. The development process of "Find Your Pawfect Buddy," a website intended to expedite animal adoption and address present issues with shelter websites, is then examined. It then goes over the approach, functionalities and schedule that would be used to create "Find Your Pawfect Buddy," a website that seeks to close the adoption gap between prospective adopters and shelters to provide stray animals with a more compassionate end.

2.2 Domain

The domain related to the "Find Your Pawfect Buddy" project is animal welfare, specifically focusing on the management and adoption of stray animals. This domain encompasses a range of activities aimed at ensuring the well-being of animals, including rescue operations, shelter management, adoption processes, and community outreach. In Malaysia, where the project is based, there is a significant issue with the stray animal population, with millions of stray dogs and cats requiring attention and care. Animal welfare organizations within this domain are dedicated to reducing the number of stray animals through various initiatives such as spaying and neutering programs, public awareness campaigns, and providing medical care. The "Find Your Pawfect Buddy" project addresses several critical aspects of this domain by creating a centralized platform for reporting stray animals, facilitating the adoption process, and

providing valuable data insights for better resource allocation. By integrating these functionalities, the project aims to improve the efficiency and effectiveness of animal welfare efforts, ultimately contributing to a decrease in stray animal populations and enhancing the overall quality of life for these animals. This domain's relevance is underscored by the growing public concern for animal rights and the need for sustainable and humane solutions to the stray animal crisis.

2.3 Existing system

A study by (Munir et al.,2023) found that approximately 6 million stray dogs and 6 million stray cats are roaming around the country. The factors mentioned as the cause of the growth of stray animal populations include ineffective containment, irresponsible breeding, and improper pet care. Moreover, according to Aseanpost, there has been a rise in animal cruelty cases from 510 in 2017 to 662 in 2018. This trend shows that there is a need for more humane treatment to address stray animals.

2.3.1 Comparison of Existing System

For the current system analysis, the website of Paws Animal Welfare Society (PAWS), a non-profit animal shelter in Petaling Jaya, has been taken as an example. In the analysis of PAWS website, there is a simple web-based system that is used to share information about donations and volunteer recruitment. However, there is still a lack of some crucial features to solve current stray animal problems. For example, if the adoptive parents want to adopt, they will need to visit the shelter physically. Also, there is a lack of reporting about stray animals to inform the shelter and save them. Lastly, there is a lack of transparency because no profiling information is shared about the stray animals.

Another study was conducted on a website in United States by the American Society for the Prevention of Cruelty to Animals (ASPCA). The website provides profile viewing for the stray cat and stray dog to be adopted. It also has a site to report animal rescue services, but it is only available via email. Next, it allows donations to be

conducted on a one-time or every-month basis. Also, there is an online shop to buy merchandise from the organization to support the stray animals.

Both websites show they have a similar agenda of receiving donations via the website as the focus. For ASPCA, profile viewing for adoptable animals is a good value added to the website compared to the website from PAWS. The facts and findings show that the Find Your Pawfect Buddy system is going to address some issues that are not matched or addressed by these systems while keeping the good elements of these websites.

2.4 Project Methodology

2.4.1 Justification

The Waterfall methodology was selected to develop the "Find Your Pawfect Buddy" website. This method works in a sequential, linear fashion, with each step being finished before going on to the next. The following justifies the use of waterfall in this project:

- a) Clearly defined requirements: The "Project Requirements" section outlines the project's objectives and functionalities. With a well-defined plan from the start, waterfall development flourishes.
- b) Focus on stability: Waterfall's structured approach reduces the chance of introducing bugs in later stages because, unlike e-commerce platforms, the website won't require frequent updates after launch.

2.4.2 Stages

- a) Requirement gathering and analysis: This first stage entails a thorough examination of the functionalities and needs of the website's users.

Developers will have a clear understanding if all requirements are documented.

- b) **System Design:** A thorough technical blueprint for the website is created based on the requirements that have been gathered. This entails deciding on suitable technologies, specifying database organization, and sketching the architecture of websites.
- c) **Development:** Using the system design document as a guide, developers create the functionalities of the website. During this stage, coding, component integration, and UI development take place.
- d) **Testing:** To find and correct errors or bugs in the functionality of the website, thorough testing is carried out. Testing usability with prospective users guarantees a user-friendly interface.
- e) **Deployment:** Public access to the final, tested website is granted upon its deployment to a live server.
- f) **Maintenance:** This stage takes care of any bug fixes or small changes that need to be made to the website, even though there may not be many updates after launch.

2.5 Project Requirements

2.5.1 Software Requirement

Visual Studio Code and MongoDBCompass.

2.5.2 Hardware Requirement

Personal laptop.

2.6 Conclusion

This chapter outlined the foundation for the selection of the Waterfall development methodology and gave a thorough analysis of pertinent literature. The project's clearly defined requirements and stability-focused approach are well-suited to the methodical approach of the Waterfall technique. The chapter also described the functional and non-functional aspects of the project, guaranteeing an easy-to-use, safe, and expandable platform for adopting animals. "Find Your Pawfect Buddy" has a clear grasp of the current situation and a well-thought-out development plan, which positions it to have a positive effect on animal welfare in Malaysia.



CHAPTER 3: ANALYSIS

3.1 Introduction

In this chapter, the project's analysis phase will be covered. In essence, analysis is the process of breaking down the entire system into smaller modules, gathering factual information, comprehending the procedures involved, spotting issues, and making workable recommendations. Therefore, the understanding of the current animal adoption process, identifying its weaknesses, and improving the new system while meeting user requirements will be the main topics of this chapter.

3.2 Current Scenario Analysis

3.2.1 Existing Scenario

The current scenario analysis of stray animal management in Malaysia reveals several shortcomings in the existing systems. Presently, stray animals are managed by a few non-profit organizations and governmental bodies, but the efforts are fragmented and lack coordination.

Current Linear Application:

- a. **Public Reporting:** When a public user encounters a stray animal, they often do not know where or how to report it efficiently. Reporting is typically done through phone calls, emails, or social media, leading to delays and missed opportunities for rescue.

- b. Shelter Response: Animal shelters receive reports from various sources but often lack a streamlined system to manage these reports. Shelters struggle with overcrowding and resource allocation due to inconsistent data on stray animal populations.
- c. Adoption Process: Potential adopters usually have to visit shelters physically to see available animals. This process is time-consuming and inconvenient, limiting the number of successful adoptions.



3.2.2 Proposed Application

Figure 3.2.2.1 shows the overall flow of process proposed application.

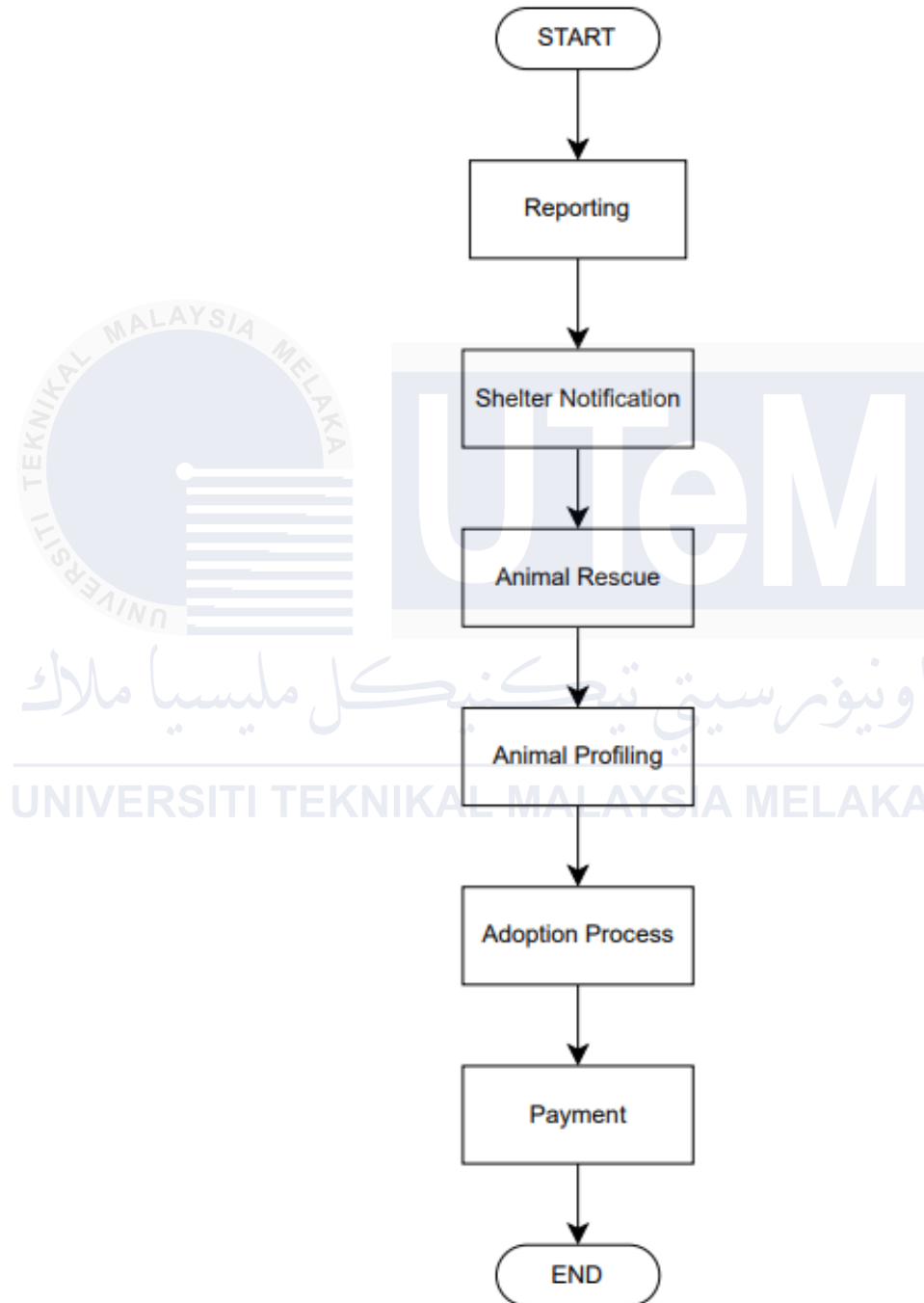


Figure 3.2.2.1 Non-Existing Application Flowchart

Figure 3.2.2.2 until figure 3.2.2.9 shows the flowchart for the module in “Find Your Pawfect Buddy” system.

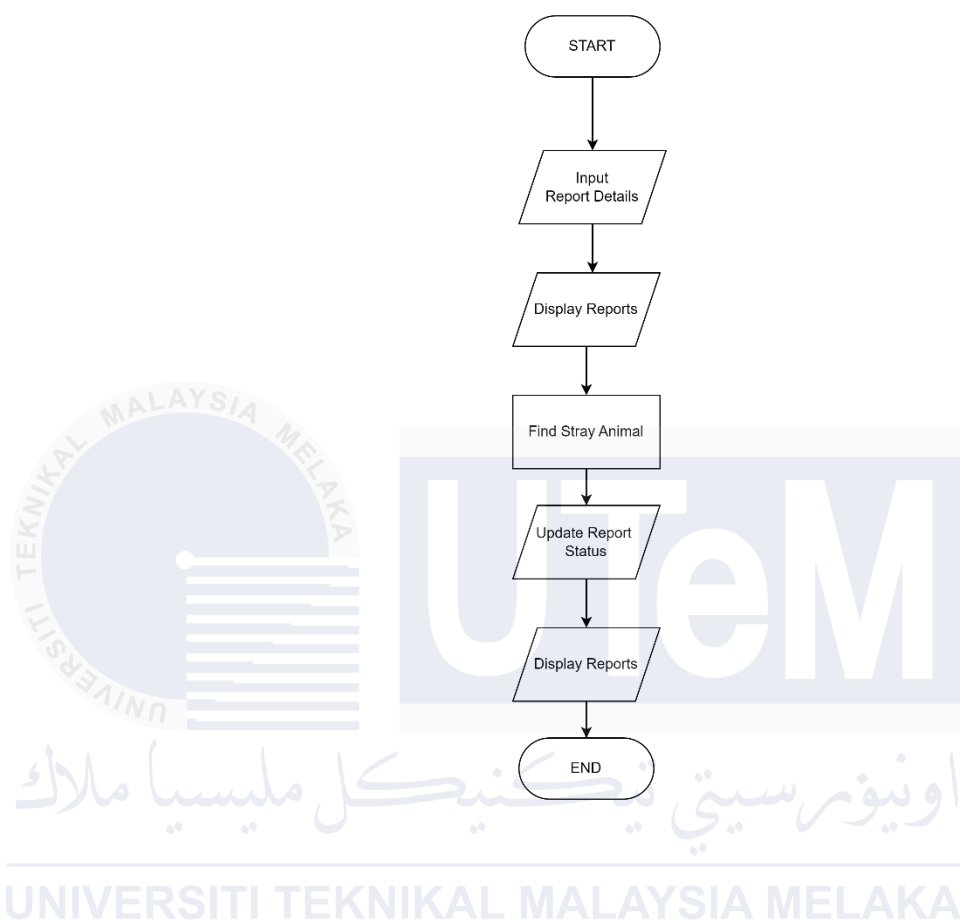


Figure 3.2.2.2 Report Animal Flowchart

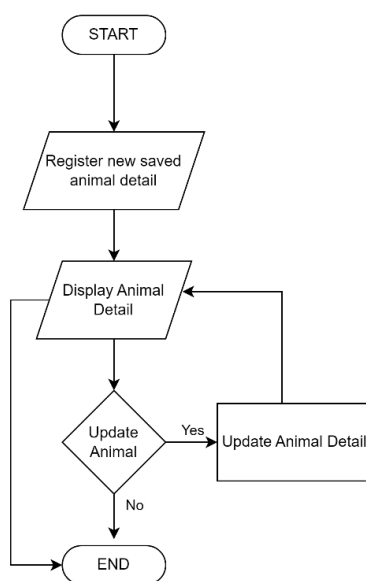


Figure 3.2.2.3 Manage Saved Animal

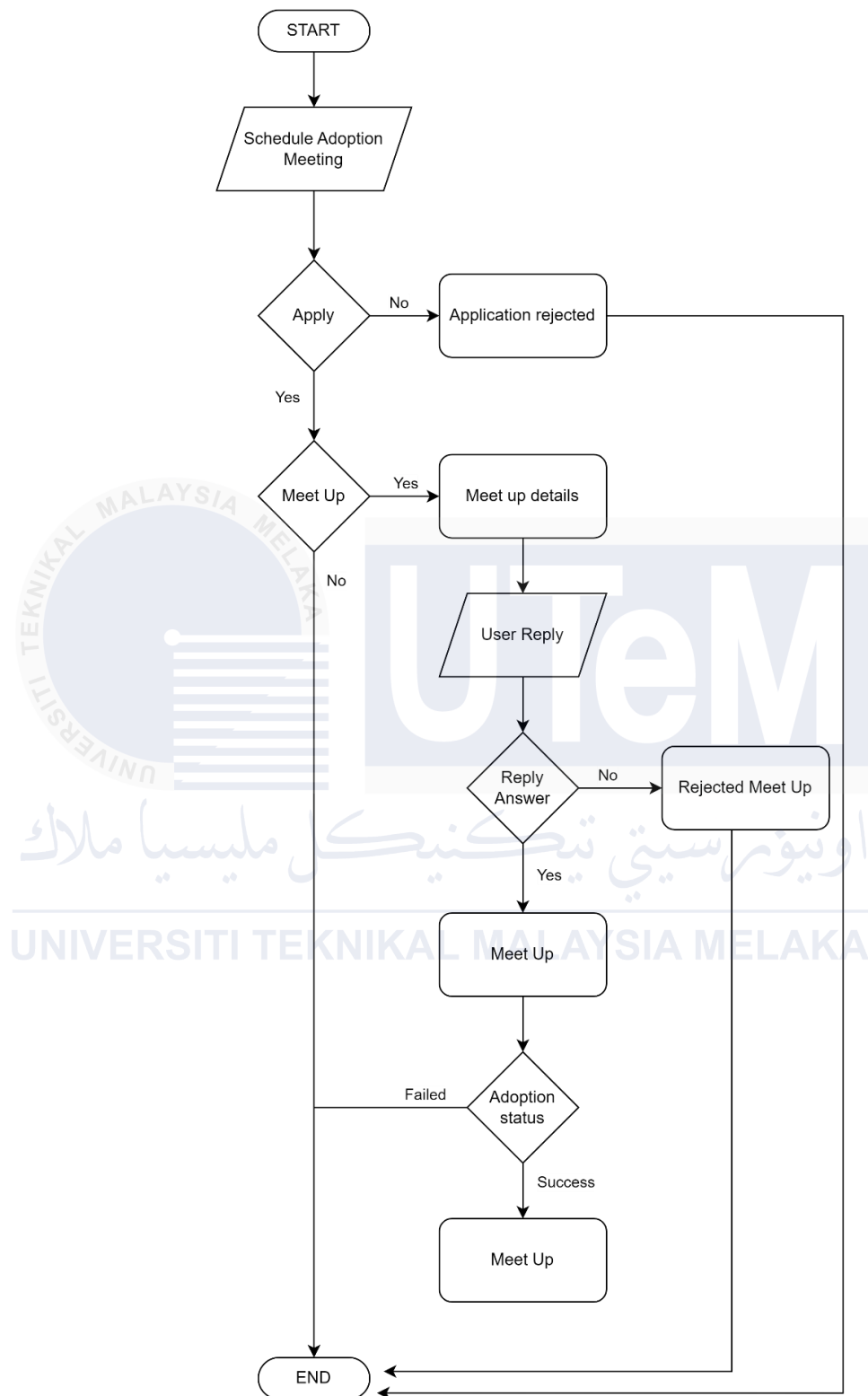


Figure 3.2.2.4 Manage Adoption Process Flowchart

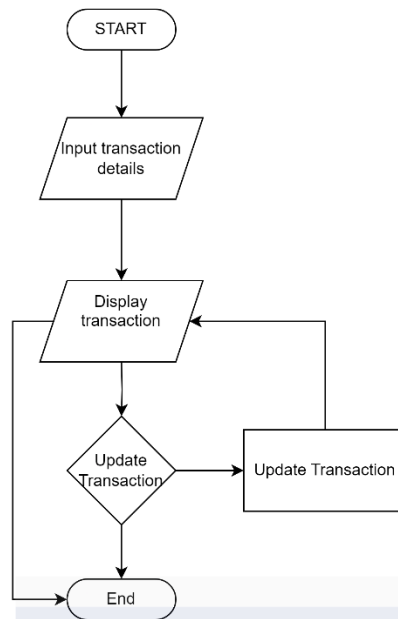


Figure 3.2.2.5 Manage Transaction Flowchart

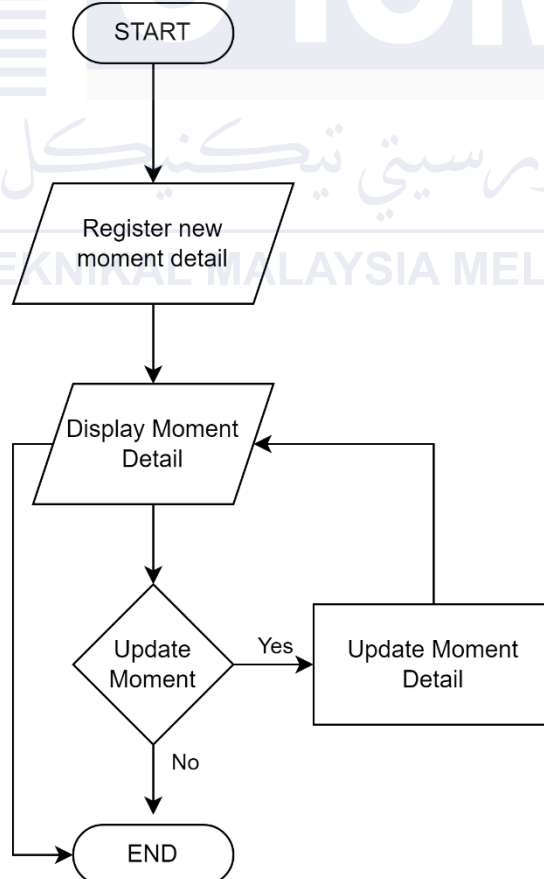


Figure 3.2.2.6 Manage Moment Flowchart

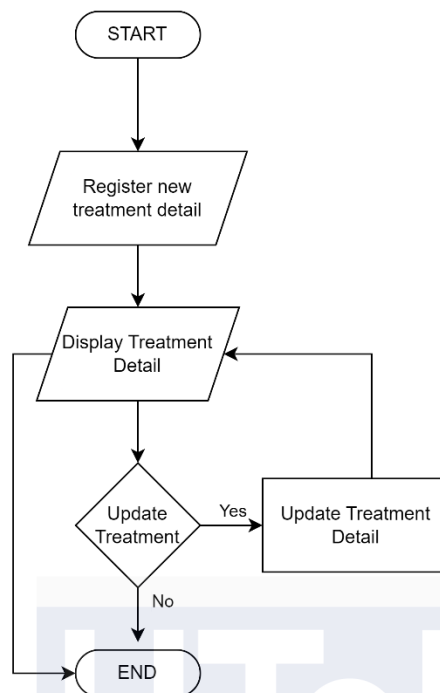


Figure 3.2.2.7 Manage Treatment Flowchart

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

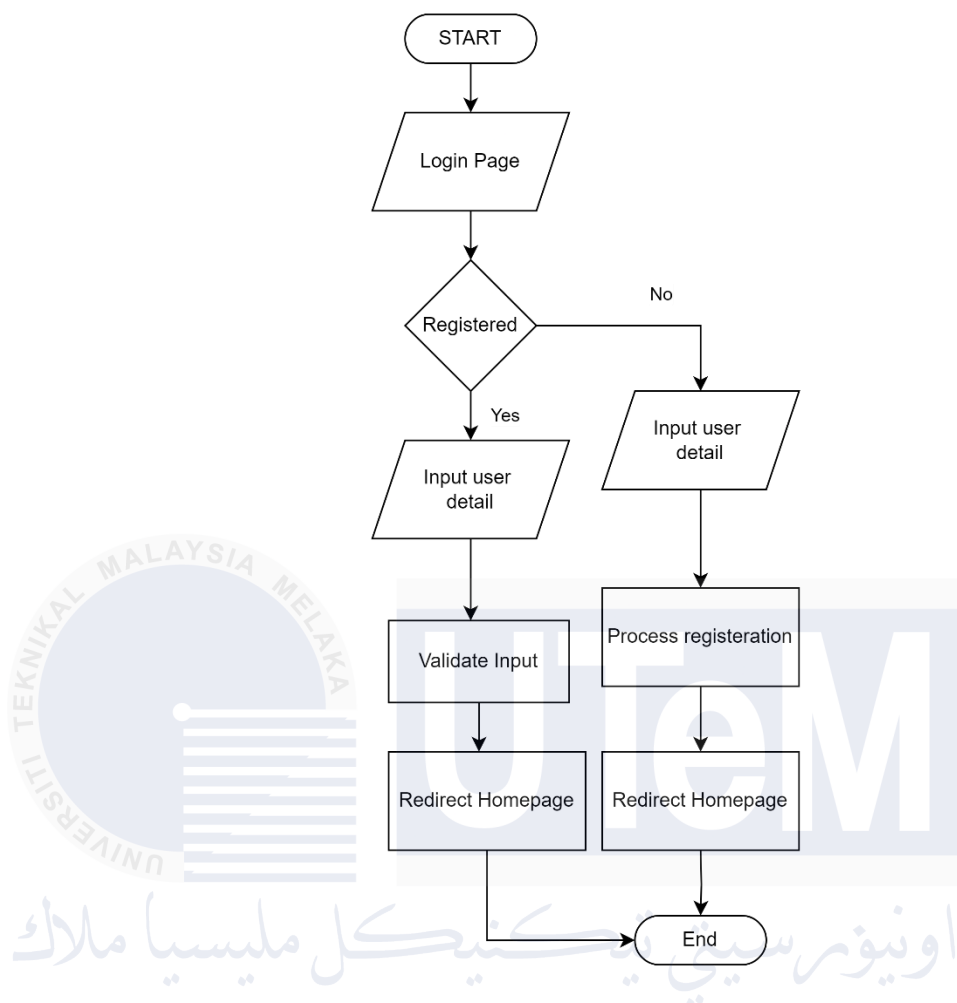


Figure 3.2.2.8 Account Management Flowchart

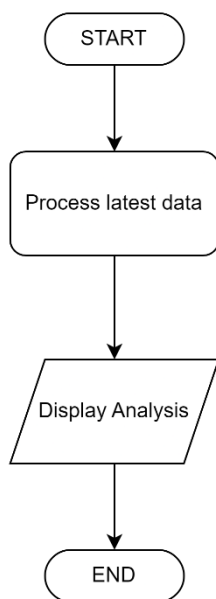


Figure 3.2.2.9 Analysis flowchart

3.3 Requirement Analysis

A thorough requirement analysis has been carried out in order to address the issues that were found. The functional and non-functional requirements needed for the "Find Your Pawfect Buddy" project's effective execution are described in this section.

3.3.1 Functional Requirement

- a) Reporting Stray Animals: Users must be able to report stray animals easily, providing pictures and location information to facilitate quick and accurate rescues.
- b) Animal Profiles: Comprehensive profiles for each animal, including photos, temperament, and breed information, must be available to potential adopters.
- c) Search and Filter: The system should allow users to search and filter adoptable animals based on various criteria such as species, age, and breed.
- d) Appointment Management: Shelter staff should be able to accept or reject adoption appointments and schedule them efficiently.
- e) Animal Treatment: Shelter staff can manage easily the animal on the platform regard animal treatment that they undergo.
- f) Animal Moment: Shelter staff can manage easily animal moments.
- g) Content Management System (CMS): Shelter staff must have the ability to quickly add, edit, and update animal profiles and other website content.
- h) Fee Management: The system should calculate and display the total adoption fees to be charged to adopters. This will be managed by staff for their record.

- i) Analysis: The system can provide an analysis based on the adoption information, report information and transaction information in visual.

3.3.2 Non-Functional Requirements

- a) Performance: The website must load quickly and adapt to users with varying internet connection speeds, ensuring a seamless user experience.
- b) Availability: The website should have high uptime, with a target of 99.5%, ensuring minimal disruption for users.
- c) Scalability: The system should handle an increasing number of users and data without performance degradation, supporting future expansion.
- d) Security: The website must protect user data, including personal information and financial transactions, through secure login procedures and robust data protection measures.

3.3.3 Software Requirement

Visual Studio Code, MongoDBCompass and XAMPP.

3.3.4 Hardware Requirement

Personal laptop.

3.4 Project Schedule and Milestones

Table 3.3.1 shows the project schedule tasks, start date and completion. Figure 3 shows the gantt chart of the visualization of the project schedule and milestones.

Tasks	Start Date	Completion
PSM1 Proposal	12/3/2024	15/3/2024
Introduction	18/3/2024	29/3/2024
Literature Review& Methodology	1/4/2024	5/4/2024
Analysis	8/4/2024	17/5/2024
Design	18/5/2024	31/5/2024
Draft Report	3/6/2024	10/6/2024
Presentation	17/6/2024	21/6/2024
Complete PSM1 Report& logbook	22/6/2024	30/6/2024

Table 3.3.4.1 Schedule Table

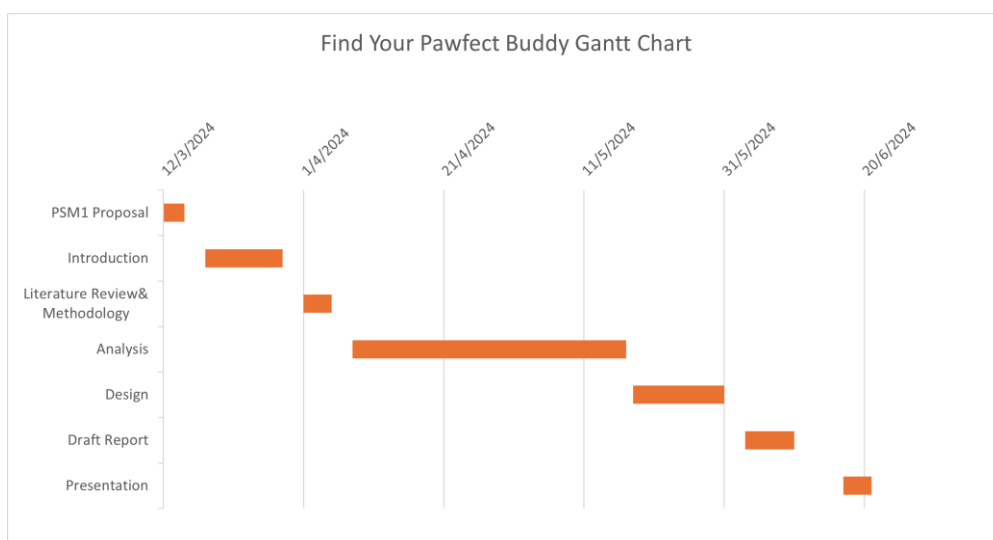


Figure 3.3.4.1 Gantt Chart

3.5 Conclusion

In this chapter, we conducted a comprehensive analysis of the current scenario, requirements, and project schedule for the "Find Your Pawfect Buddy" initiative. The analysis revealed significant inefficiencies in Malaysia's stray animal management, highlighting the fragmented efforts of non-profit organizations and governmental bodies, inefficient reporting systems, overcrowded shelters, and a cumbersome adoption process. To address these issues, we established functional requirements for an easy-to-use reporting system, comprehensive animal profiles, advanced search and filter options, an online adoption application, and efficient appointment management. Non-functional requirements focused on performance, availability, scalability, and security. The detailed project schedule provides a clear roadmap from proposal to final report completion. This chapter lays the groundwork for developing a cohesive, efficient, and user-friendly system to improve stray animal management and adoption processes in Malaysia, aiming for a significant positive impact on animal welfare and the adoption experience.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter outlines the design approach for our animal adoption website. The primary objectives of our design are to create an intuitive, visually appealing interface that facilitates easy browsing and encourages animal adoption and simplify the animal management. Also, the requirement of the application in term of database will be show in this chapter.

4.2 System Architecture Design

4.2.1 Scene Sequence Diagram

Figure 4.2.1 illustrates the sequence for reporting stray animals, managing animal profiles, and facilitating adoptions. Users report strays with pictures and locations, which the system processes to update animal profiles. Potential adopters can search and filter these profiles, apply online, and schedule appointments. Shelter staff manage appointments, update content via a CMS, and the system calculates adoption fees.



Figure 4.2.1.1 Sequence Diagram

4.2.2 High Level Design

For normal viewing, the website consists of five main sections: Home, Browse, Report, Staff and User. The layout features a responsive design with a fixed header for easy navigation. Figure 4.2.2.1 shows the user wireframes which consists of home, browse and report page.

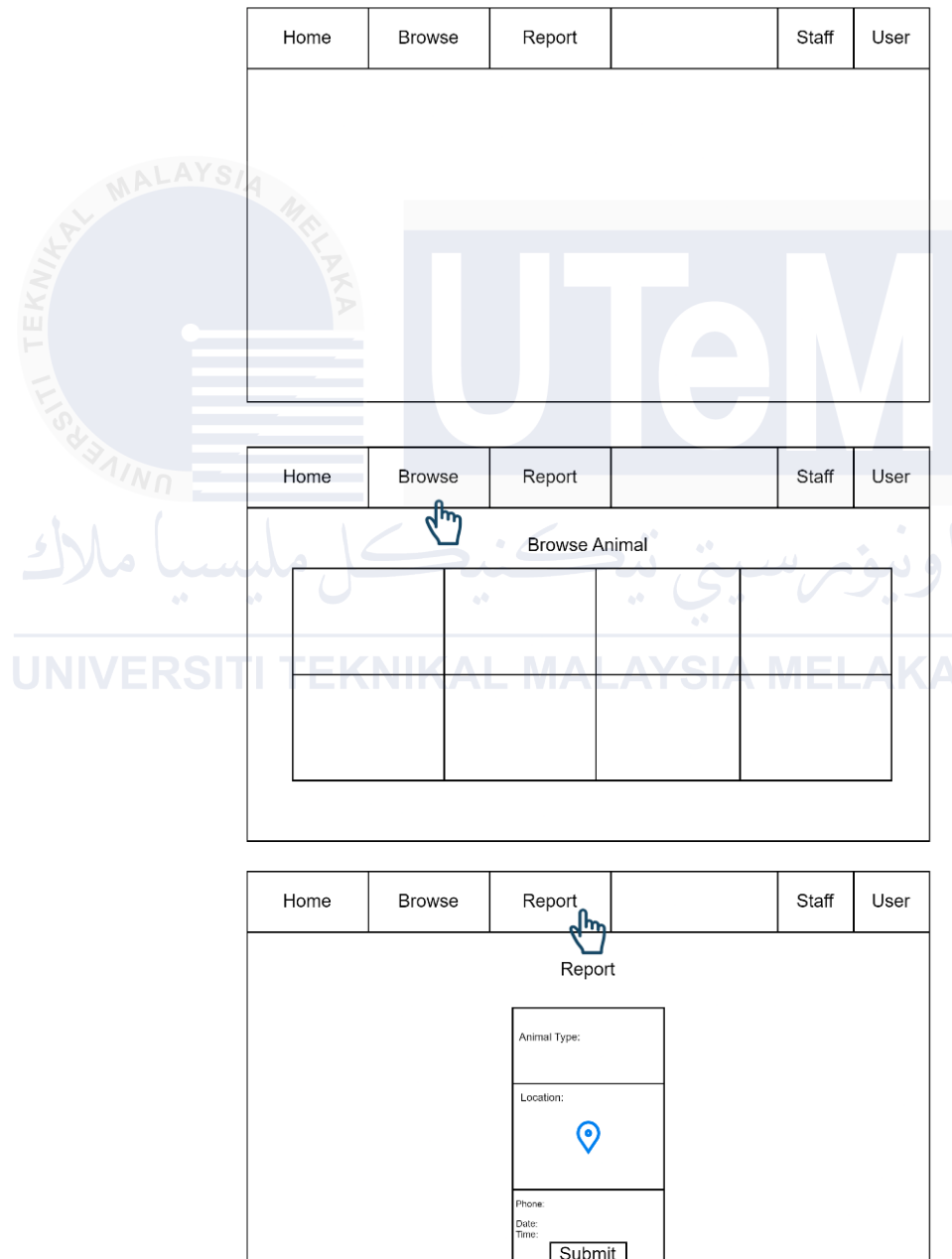


Figure 4.2.2.1 User Wireframes

Figure 4.2.2.2 shows the staff wireframe which consists of latest report, staff management, adoption and summary page.

The figure displays four wireframe screenshots of a staff management system, each showing a navigation bar with the following items: Home, Latest Report, Staff Management, Adoption, Summary, and Logout.

Wireframe 1: Shows the navigation bar and a large empty content area.

Wireframe 2: Shows the navigation bar with a hand cursor pointing to 'Latest Report'. Below the navigation bar is a 'Search:' input field and a table with 3 rows and 3 columns.

Wireframe 3: Shows the navigation bar with a hand cursor pointing to 'Staff Management'. Below the navigation bar is an 'Update Detail:' section with the following input fields:

- Staff Id:
- Old Password:
- New Password:

Wireframe 4: Shows the navigation bar with a hand cursor pointing to 'Adoption'. Below the navigation bar is a 'Search:' input field and a table with 3 rows and 3 columns.

Wireframe 5: Shows the navigation bar with a hand cursor pointing to 'Summary'. Below the navigation bar is a 'Summary' section with the following labels:

- Reported stray animals:
- Saved stray animals:
- Adopted animals:
- Total transaction amount:

Below these labels is a box containing three location pins.

Figure 4.2.2.2 Staff Wireframes

4.2.3 Technology Stack

The website uses HTML, CSS, and JavaScript for the frontend, Python with Flask for the backend, and MongoDB for the database. These technologies were chosen for their robustness, scalability, and ease of use.

4.3 Database Design

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and mostly correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationship.

4.3.1 Conceptual Design

1. Context diagram

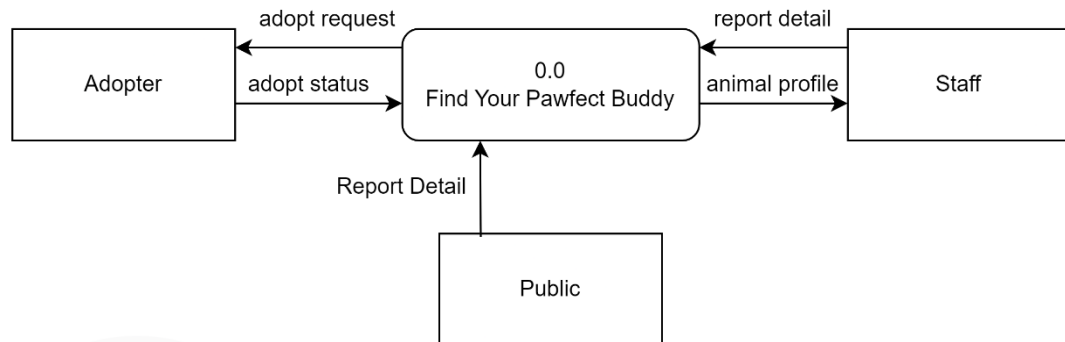


Figure 4.3.1.1 DFD Level 0

2. Data Flow Diagram Level 1

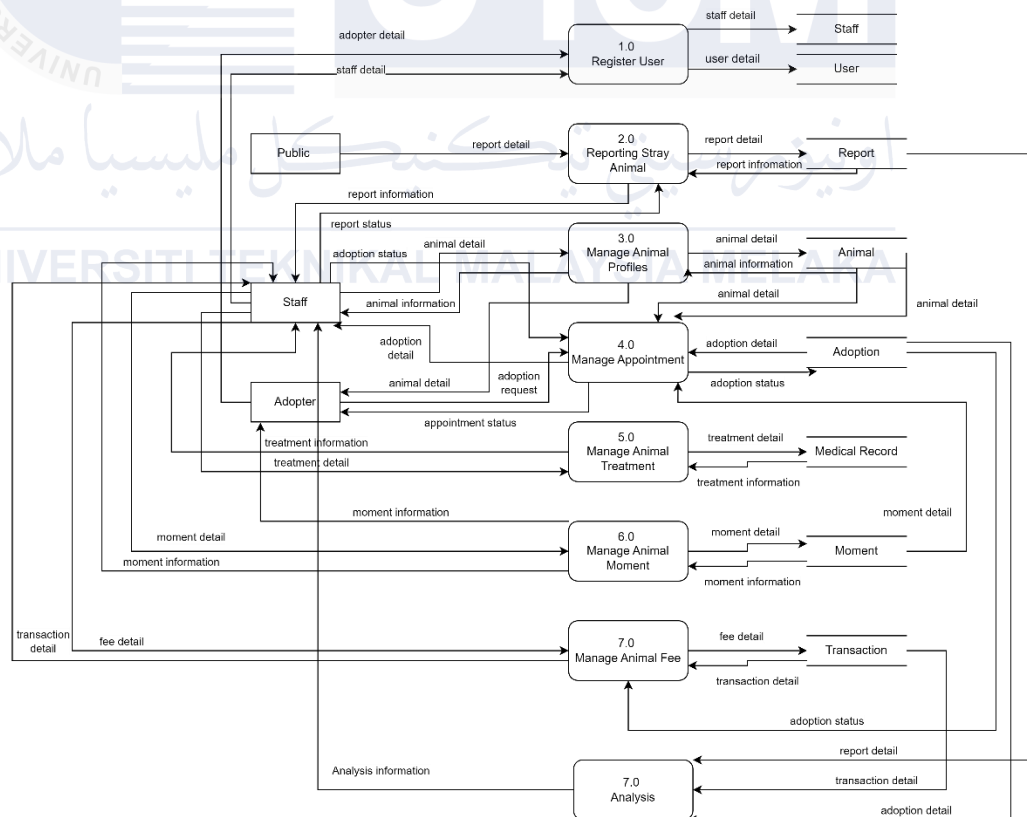


Figure 4.3.1.2 DFD Level 1

3. Data Flow Diagram Level 2

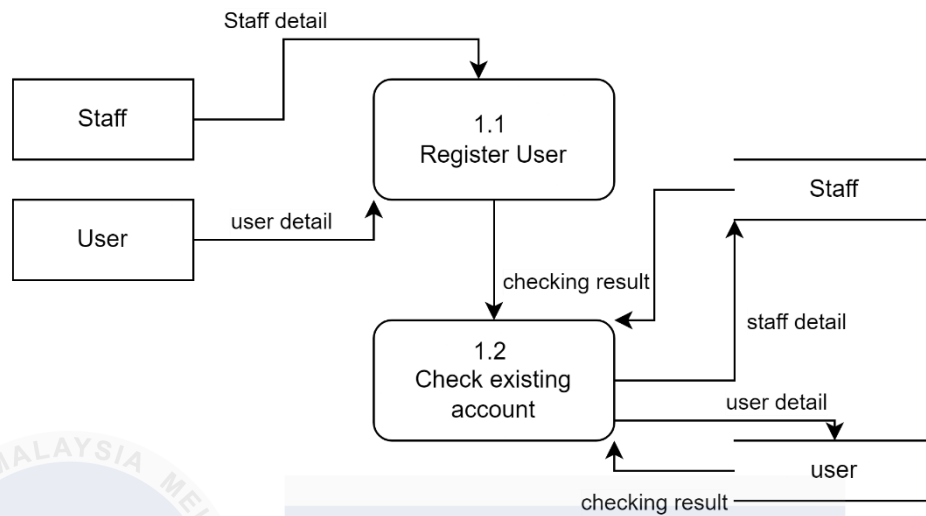


Figure 4.3.1.3 DFD Level 2 Register User

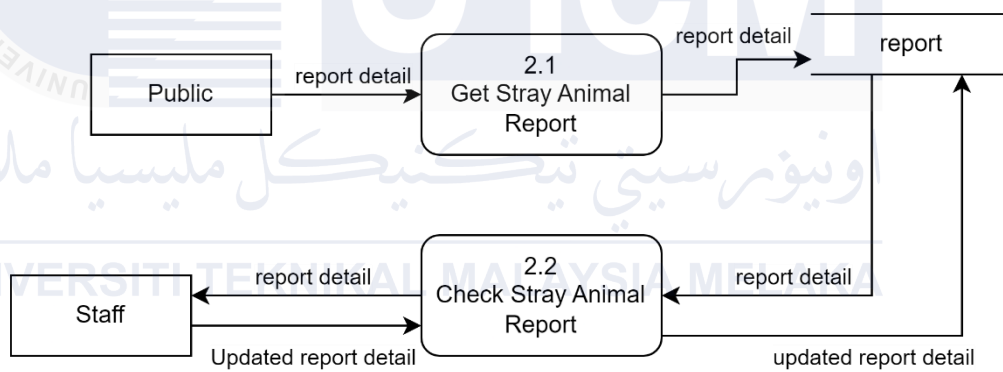


Figure 4.3.1.1 DFD Level 2 Report Stray Animal

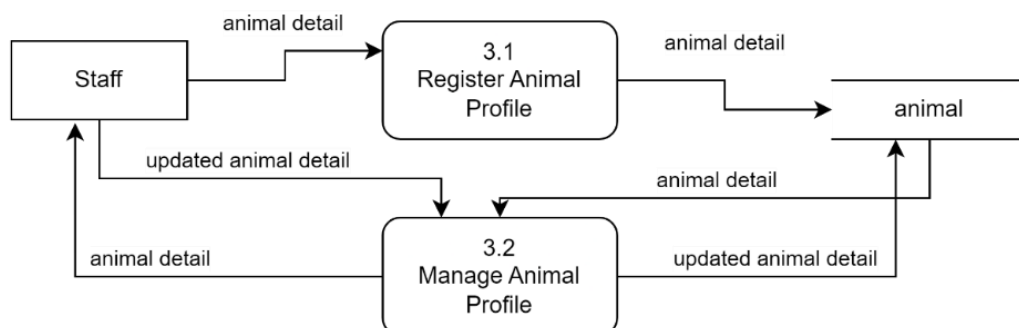


Figure 4.3.1.2 DFD Level 2 Animal Profile

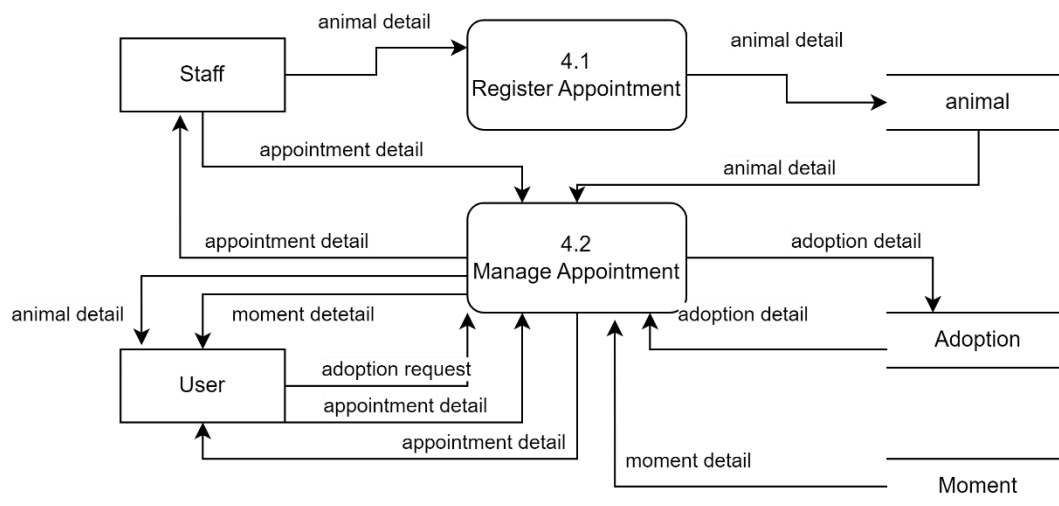


Figure 4.3.1.3 DFD Level 2 Manage Appointment

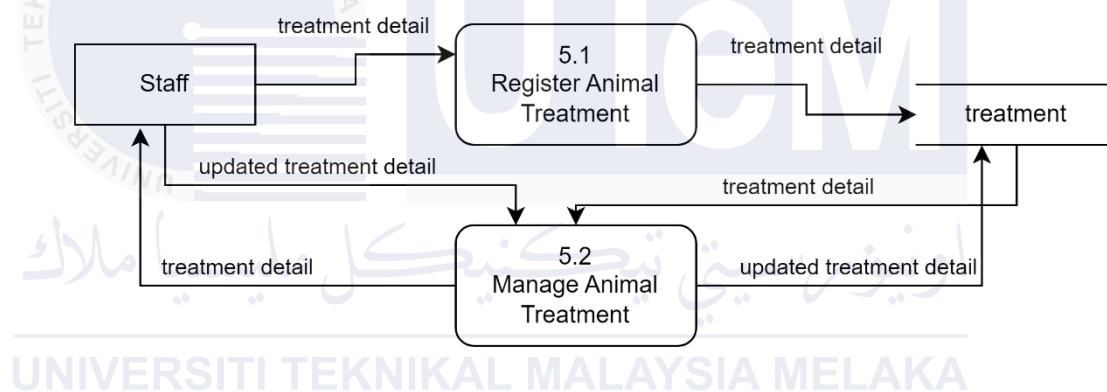


Figure 4.3.1.4 DFD Level 2 Manage Treatment

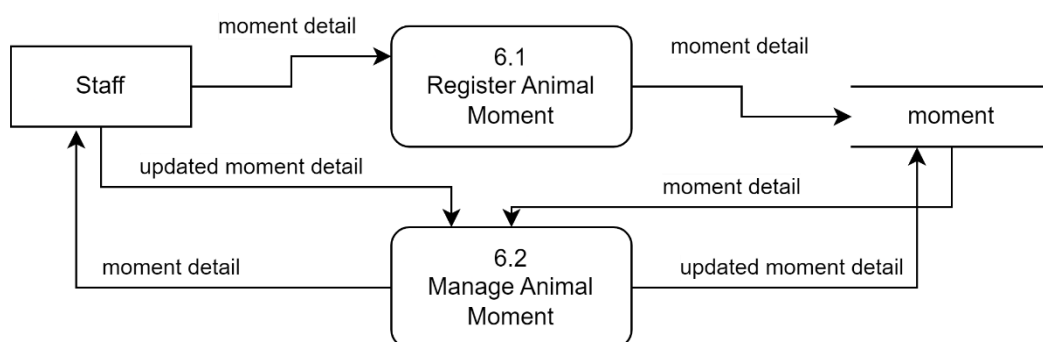


Figure 4.3.1.5 DFD Level 2 Manage Animal Moment

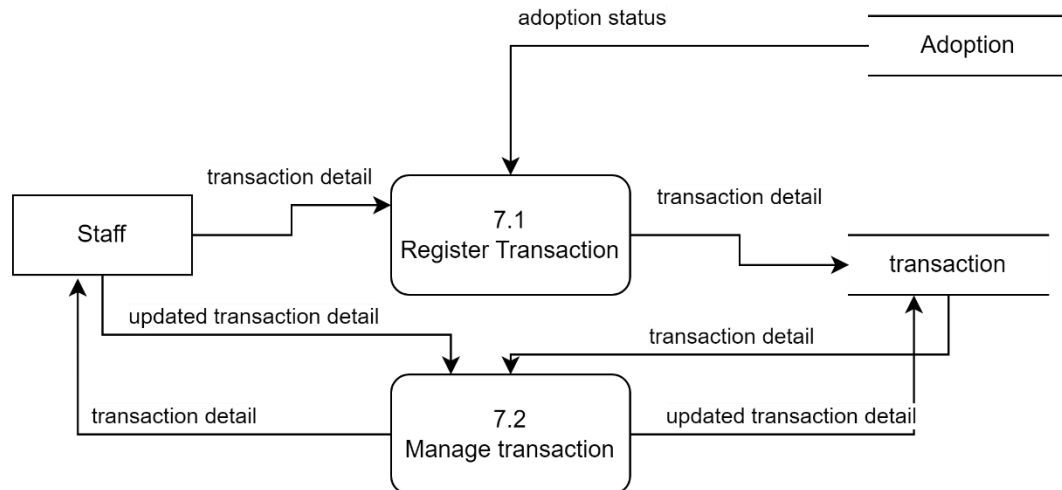


Figure 4.3.1.6 DFD Level 2 Manage Fee

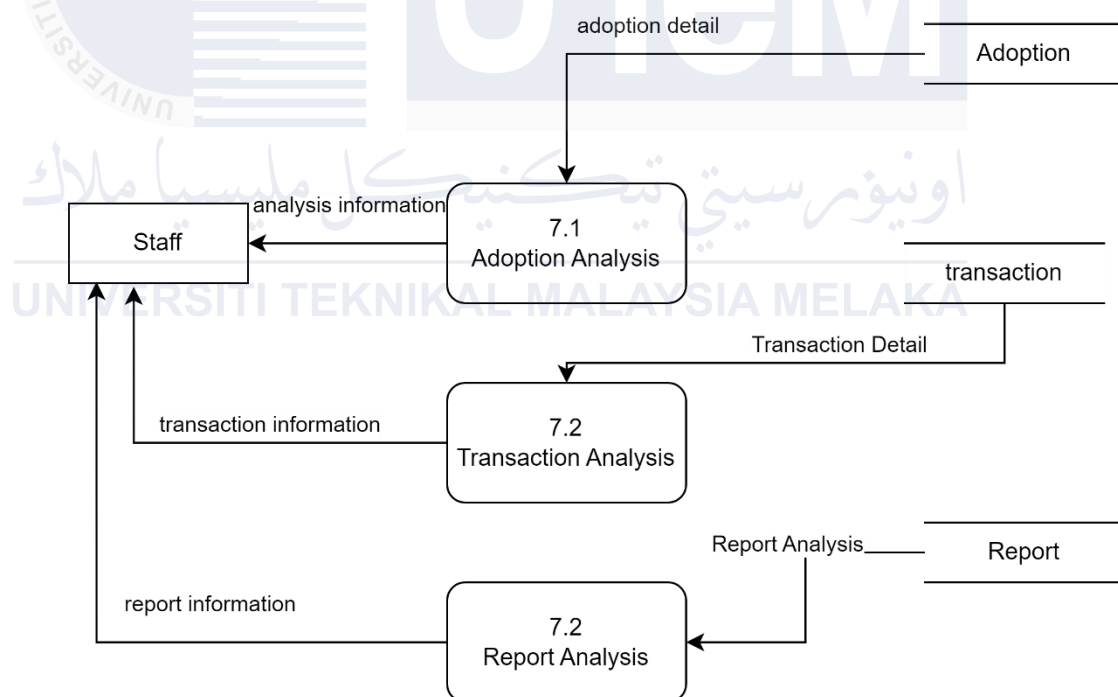


Figure 4.3.1.7 DFD Level 2 Analysis

4.3.2 Logical Design

While NoSQL databases like MongoDB offer schema flexibility, starting with a conceptual design remains valuable for understanding data relationships and overall structure. For the initial phase of our database design, we utilized an Entity-Relationship Diagram (ERD) to model the core entities and their relationships in our "Find Your Pawfect Buddy" application. Figure 4.3.1.1 illustrates the Entity-Relationship Diagram for "Find Your Pawfect Buddy". This ERD serves as a conceptual foundation, guiding our MongoDB schema design decisions. It helps visualize the key entities such as animal, user, adoption and more and for their interconnections, even though the actual implementation in MongoDB will be more flexible and potentially denormalized.

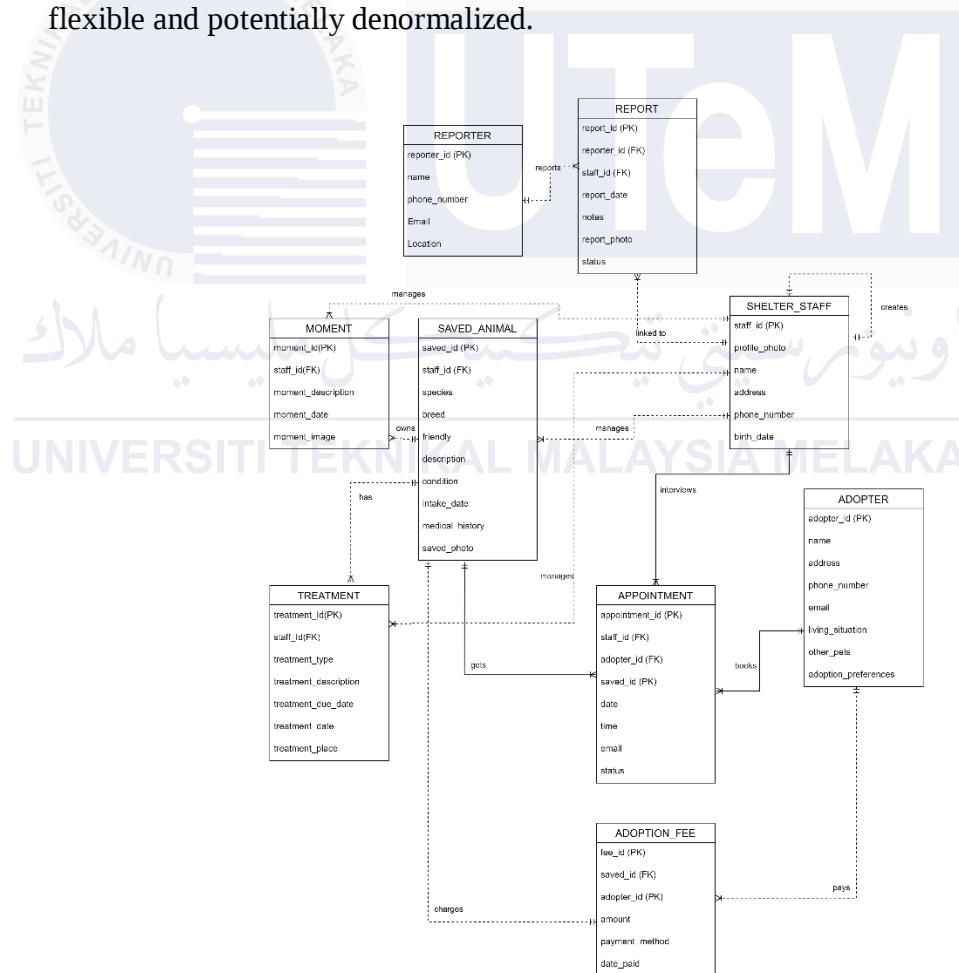


Figure 4.3.2.1 System ERD

4.3.3 Data Model

Our database consists of seven main collections:

- a) Animal
- b) Adoption
- c) Medical Record
- d) Moment
- e) Report
- f) Staff
- g) Transaction
- h) User



4.3.4 Collection Structures

a) Animal Collection

- Stores detailed information about each animal
- Key fields: animal_id (unique identifier), species, breed, age, health details, and adoption status
- Includes binary data for animal photos

b) Adoption Collection

- Records adoption applications and their status
- Links to both animals and users
- Key fields: adoption_id, animal_id, username, adoption_status

c) Medical Collection

- Tracks medical treatments and check-ups for animals
- Linked to animals via animal_id
- Key fields: treatment_id, animal_id, treatment_date, treatment_type

d) Moment Collection

- Stores special moments or updates about animals
- Linked to animals via animal_id
- Includes binary data for moment photos

e) Report Collection

- Manages reports of animals needing rescue
- Key fields: report_id, animal_type, location, report_status

f) Staff Collection

- Stores information about shelter staff
- Key fields: staffid, password (hashed)

g) Transaction Collection

- Records financial transactions related to adoptions
- Linked to adoptions and animals
- Key fields: transaction_id, adoption_id, animal_id, amount

h) User Collection

- Stores information about registered users (potential adopters)
- Key fields: username, password (hashed), contact information

4.3.5 Relationships and data linking

- Animals are central to the database, linked to adoptions, medical records, moments, and transactions via animal_id
- Users are linked to adoptions and transactions via username
- Adoptions link animals, users, and transactions

4.3.6 Key Design Decisions

a) Use of Unique Identifiers: Each collection has a unique identifier such as animal_id and adoption_id for easy reference and linking.

b) Embedded vs. Referenced Data: We've chosen to use references such as animal_id in adoptions rather than embedding to maintain data consistency and allow for easier updates.

c) Binary Data Storage: Photos are stored as binary data directly in the animals and moments collections, allowing for quick retrieval with animal information.

d) Flexible Schema: The schema allows for additional fields to be added easily, such as new health indicators for animals.

e) Security: Passwords in the staff and user collections are stored as hashed values for security.

4.4 Graphical User Interface (GUI) Design

A Graphical User Interface design is a crucial component that defines user interaction with a system, encompassing visual elements and interactive features that facilitate engagement and functionality. For "Find Your Pawfect Buddy," the GUI design focuses on creating an intuitive, efficient, and user-friendly interface to enhance the overall user experience. It comprises three key components: Navigation Design, which creates a logical structure for users to move through the application; Input Design,

which involves user-friendly forms and interactive elements for data entry and selections; and Output Design, which focuses on how information is presented to users, including animal profiles and search results. These components are carefully crafted to ensure a seamless and engaging user experience, facilitating efficient navigation, data entry, and information retrieval within the animal adoption website. The subsequent sections will explore the specific design choices and rationale for each of these GUI components in "Find Your Pawfect Buddy."

4.4.1 Navigation Design

The navigation component of the interface design provides a straightforward access and clearly direction for the user. Figure 4.4.1.1 shows the navigation path of the user and staff when they are access the “Find Your Pawfect Buddy”.

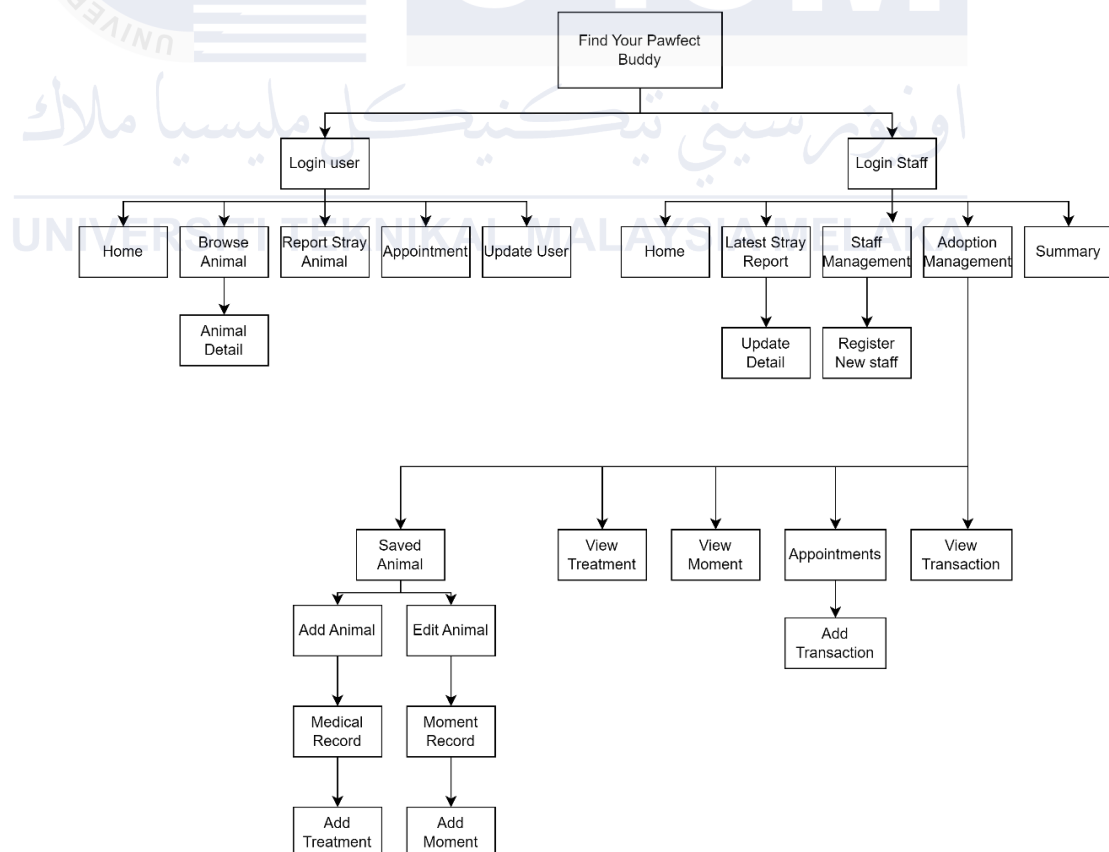
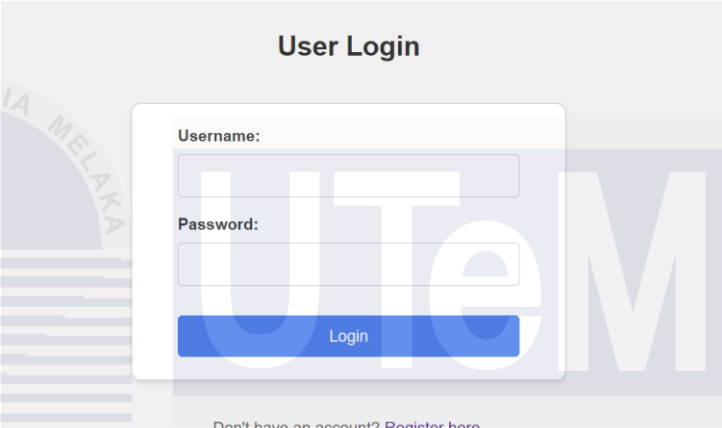


Figure 4.4.1.1 Navigation Paths

4.4.2 Input Design

The input design focuses on the entry of data into the system by users in the form of structured and unstructured data. The screen and forms are designed and used to store information for the system for an action performed. The Figure 4.4.2.1 until Figure 4.4.2.14 illustrates the input design.

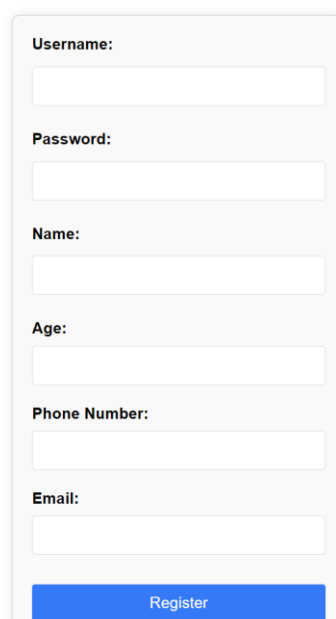
Figure 4.4.2.1 User Login



The figure shows a 'User Login' form. It has a title 'User Login' at the top. Below the title, there are two input fields: 'Username:' and 'Password:'. Each field has a corresponding text input box. Below these fields is a blue button labeled 'Login'. At the bottom of the form, there is a link that says 'Don't have an account? [Register here.](#)'. The background of the form is light gray, and there is a large, semi-transparent watermark of the 'UTeM' logo and the text 'UNIVERSITI TEKNIKAL MALAYSIA MELAKA' and 'اونيورسيتي تيكنيكل مليسيا ملاك'.

Figure 4.4.2.2 User Register

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



The figure shows a 'User Register' form. It has a title 'Register' at the top. Below the title, there are six input fields: 'Username:', 'Password:', 'Name:', 'Age:', 'Phone Number:', and 'Email:'. Each field has a corresponding text input box. At the bottom of the form is a blue button labeled 'Register'. The background of the form is light gray, and there is a large, semi-transparent watermark of the 'UTeM' logo and the text 'UNIVERSITI TEKNIKAL MALAYSIA MELAKA' and 'اونيورسيتي تيكنيكل مليسيا ملاك'.

Figure 4.4.2.3 Schedule Appointment

Name: Dog2
Species: dog
Breed: German Shepherd

Schedule Adoption Meeting

Suggested Meet-up Date/Time 1:

Suggested Meet-up Date/Time 2:

Suggested Meet-up Date/Time 3:

Submit Suggestions

short.

Moment Record

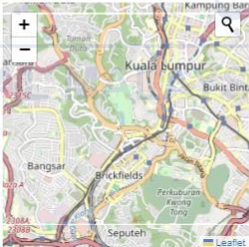
No animal moments recorded.

Schedule Adoption Meeting

Figure 4.4.2.4 Report Stray Animal

Report Stray Animal

Animal Type:
Choose One

Location:


Description:

Phone Number:
Exp: 0112312312

Date:
15 / 08 / 2024

Time:
04 : 01 PM

Photo:
 No file chosen

Submit Report

Figure 4.4.2.5 Appointment Information

Appointment Information

Adoption ID	A08e9
Animal ID	A6099
Animal Name	dog grey
Adoption Status	apply
Submission Date and Time	2024-08-05 14:01:26.086000

Adoption ID	Ae5a9
Animal ID	A59c7
Animal Name	Cat2
Adoption Status	apply
Submission Date and Time	2024-08-03 21:53:57.683000

Adoption ID	Ada1a
Animal ID	A6099
Animal Name	dog grey
Adoption Status	meet_up
Meeting Location	
Submission Date and Time	2024-07-16 14:14:19.862000

Accept Meet Up
Reject Meet Up

Figure 4.4.2.6 User Update Details

Update Details

Username:
a

Old Password (required for any changes):

New Password:

Confirm New Password:

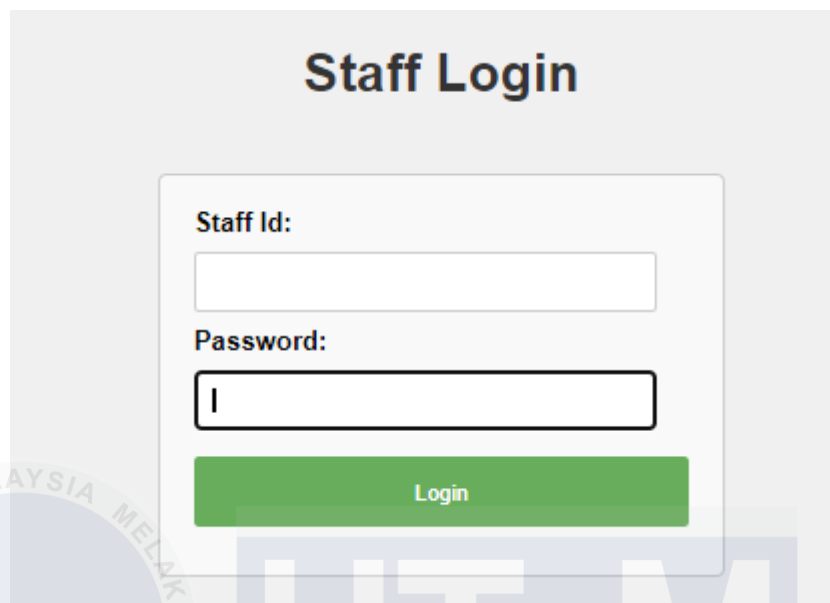
Phone:
017289212

Email:
asim2@gmail.com

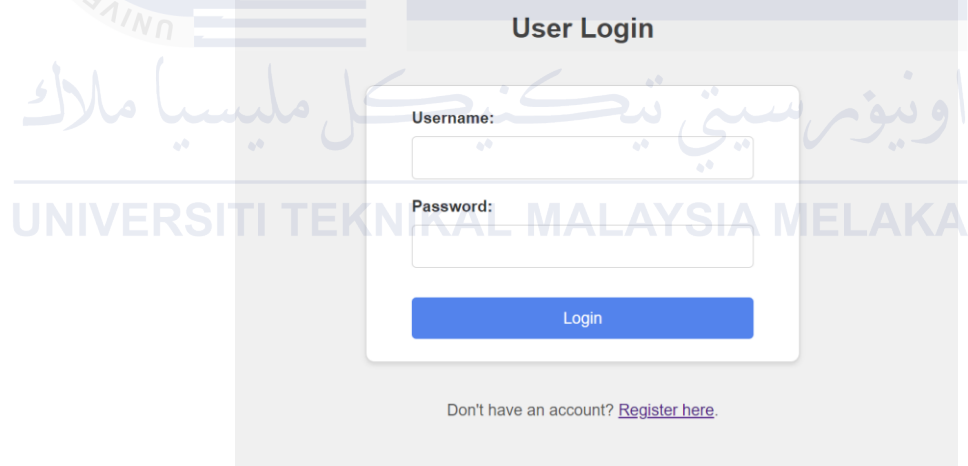
Name:
12

Age:
22

Update
Delete User

Figure 4.4.2.7 Staff Login

The Staff Login form is titled "Staff Login" in a large, bold, black font. It contains two input fields: "Staff Id:" and "Password:". The "Staff Id:" field is a white rectangle with a thin black border. The "Password:" field is a white rectangle with a thin black border and a small black cursor icon. Below the password field is a green button with the text "Login" in white. The form is set against a light gray background.

Figure 4.4.2.8 User Login

The User Login form is titled "User Login" in a large, bold, black font. It contains two input fields: "Username:" and "Password:". The "Username:" field is a white rectangle with a thin black border. The "Password:" field is a white rectangle with a thin black border. Below the password field is a blue button with the text "Login" in white. At the bottom of the form, there is a link: "Don't have an account? [Register here.](#)". The form is set against a light gray background. There is a large, semi-transparent watermark in the background that reads "UTeM" and "UNIVERSITI TEKNIKAL MALAYSIA MELAKA" in both English and Malay.

Figure 4.4.2.9 User Register

Register

Username:

Password:

Name:

Age:

Phone Number:

Email:

[Register](#)

Figure 4.4.2.10 Schedule Appointment

Name: Dog2
 Species: dog
 Breed: German Shepherd

Schedule Adoption Meeting x

Suggested Meet-up Date/Time 1:

Suggested Meet-up Date/Time 2:

Suggested Meet-up Date/Time 3:

[Submit Suggestions](#)

Moment Record

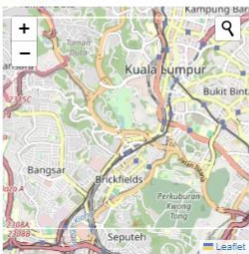
No animal moments recorded.

[Schedule Adoption Meeting](#)

Figure 4.4.2.11 Report Stray Animal

Report Stray Animal

Animal Type:
Choose One ▼

Location:


Description:

Phone Number:
Exp: 0112312312

Date:
15/08/2024

Time:
04:01 PM

Photo:
Choose File | No file chosen

Submit Report

Figure 4.4.2.12 Appointment Information

Appointment Information	
Adoption ID	A08e9
Animal ID	A6099
Animal Name	dog grey
Adoption Status	apply
Submission Date and Time	2024-08-05 14:01:26.086000

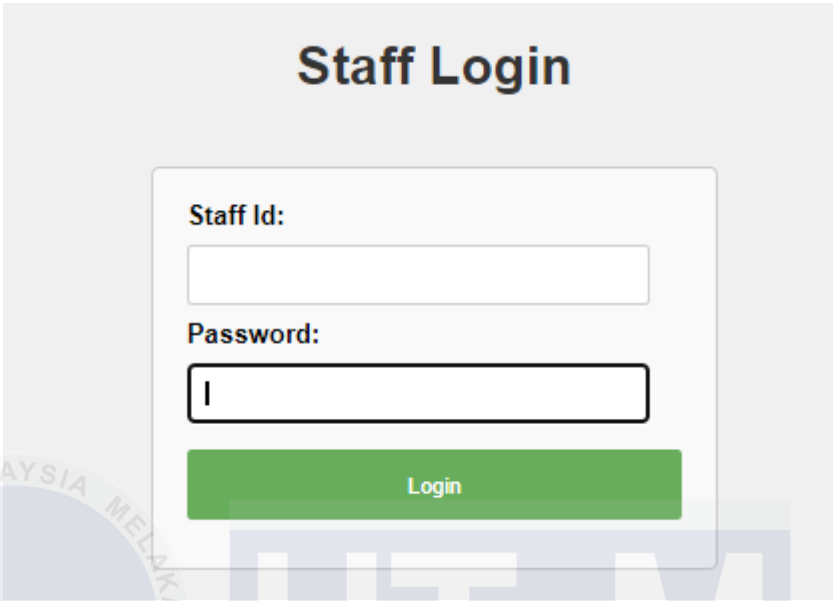
Adoption ID	Ae5a9
Animal ID	A59c7
Animal Name	Cat2
Adoption Status	apply
Submission Date and Time	2024-08-03 21:53:57.683000

Adoption ID	Ada1a
Animal ID	A6099
Animal Name	dog grey
Adoption Status	meet_up
Meeting Location	
Submission Date and Time	2024-07-16 14:14:19.862000

Figure 4.4.2.13 User Update Details

Update Details
Username: a
Old Password (required for any changes):
New Password:
Confirm New Password:
Phone: 017289212
Email: asim2@gmail.com
Name: 12
Age: 22
Update
Delete User

Figure 4.4.2.14 Staff Login



Staff Login

Staff Id:

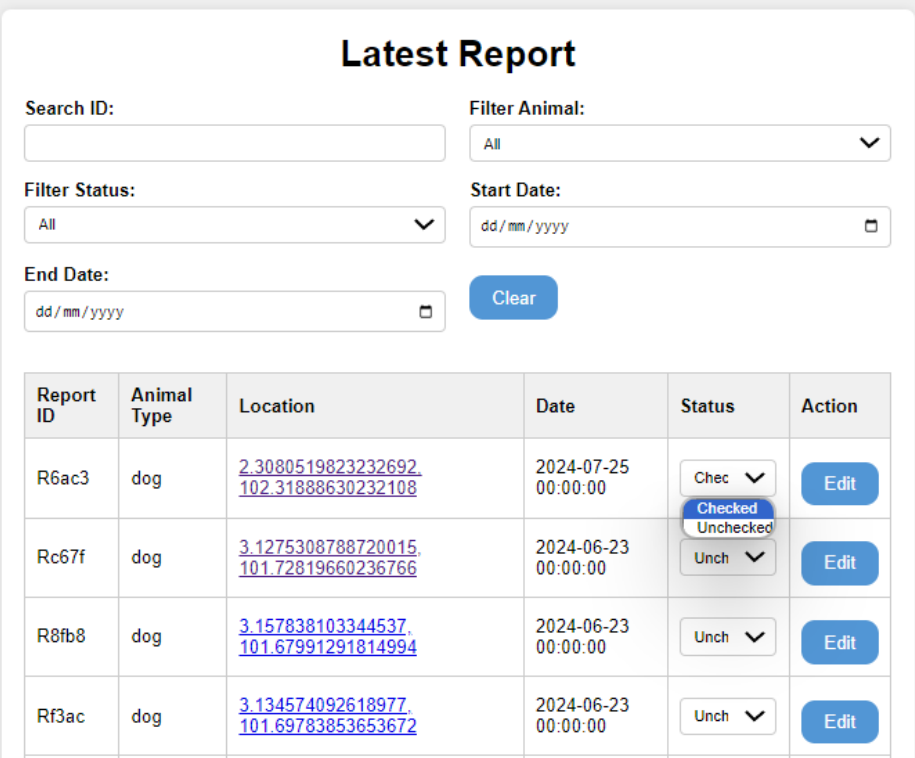
Password:

Login

4.4.3 Output Design

The output design concerns on presenting retrieved information of the system on the screen or form. The output design was showed at Figure 4.4.3.1 until Figure 4.4.3.40.

Figure 4.4.3.1 Update Report Status



Latest Report

Search ID:

Filter Animal:

Filter Status:

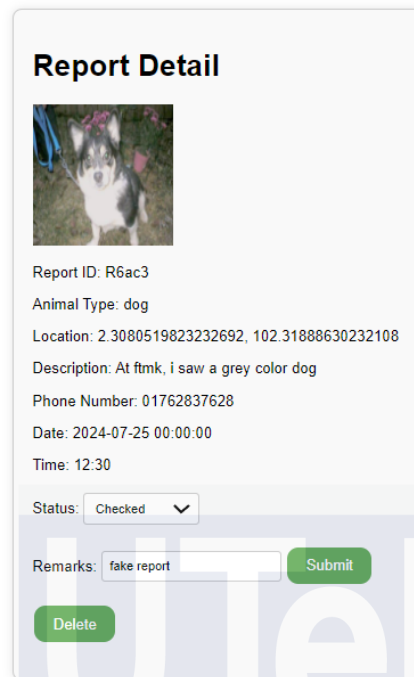
Start Date:

End Date:

Clear

Report ID	Animal Type	Location	Date	Status	Action
R6ac3	dog	2.3080519823232692.102.31888630232108	2024-07-25 00:00:00	Chec <input checked="" type="text" value="Checked"/> Unch	Edit
Rc67f	dog	3.1275308788720015.101.72819660236766	2024-06-23 00:00:00	Unch Unch	Edit
R8fb8	dog	3.157838103344537.101.67991291814994	2024-06-23 00:00:00	Unch	Edit
Rf3ac	dog	3.134574092618977.101.69783853653672	2024-06-23 00:00:00	Unch	Edit

Figure 4.4.3.2 Report Detail



Report Detail

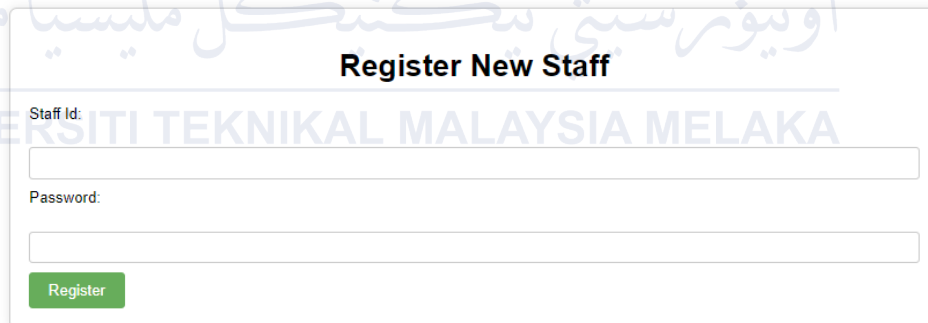


Report ID: R6ac3
Animal Type: dog
Location: 2.3080519823232692, 102.31888630232108
Description: At fmk, i saw a grey color dog
Phone Number: 01762837628
Date: 2024-07-25 00:00:00
Time: 12:30

Status: ☒ Checked

Remarks:

Figure 4.4.3.3 Register New Staff



Register New Staff

Staff Id:

Password:

Figure 4.4.3.4 Update Staff Detail

Update Your Details

Staff ID:

Old Password (required for any changes):

New Password:

Confirm New Password:

Phone:

Email:

Address:

Birthdate:

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

اونیورسیتی تکنیکل ملیسیا ملاک

UTeM

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.4.3.5 Animal Profile Update



The form is titled "Animal Profile Update" and features a header image of a black and tan dog. Below the header, the form contains several sections for updating animal information. The "New Photo" section has a "Choose File" button and a "No file chosen" status. The "Name" field is filled with "dog grey". The "Species" dropdown is set to "Dog". The "Breed" field is filled with "chia". The "Age" field is filled with "5". The "Overall Health" dropdown is set to "Good". The "Vaccination" section has two radio buttons: "Up-to-date" (selected) and "Needs Vaccinations". The "Spayed/Neutered" section has two radio buttons: "Yes" (selected) and "No". The "Energy Level" dropdown is set to "Medium". The "Temperament" dropdown is set to "Anxious". The "Trainability" dropdown is set to "Independent". The "Interaction with Other Animals" section has three checkboxes: "Good with Dogs" (checked), "Good with Cats" (unchecked), and "Good with Children" (checked). The "Date of Intake" field is filled with "25/06/2024". The "Last Grooming Date" field is filled with "11/06/2024". The "Notes" field is filled with "he is easily hungry". The "Description" field is filled with "an adorable dog that will make you happy!". At the bottom of the form, there are four blue buttons: "Update Profile", "Delete Profile", "Medical Record", and "Moment Record".

Animal Profile Update

Animal Id: A8099

New Photo:

[Choose File](#) No file chosen

Name:

dog grey

Species:

Dog

Breed:

chia

Age:

5

Overall Health:

Good

Vaccination:

☒ Up-to-date ☐ Needs Vaccinations

Spayed/Neutered:

☒ Yes ☐ No

Energy Level:

Medium

Temperament:

Anxious

Trainability:

Independent

Interaction with Other Animals:

☒ Good with Dogs ☐ Good with Cats ☒ Good with Children

Date of Intake:

25/06/2024

Last Grooming Date:

11/06/2024

Notes:

he is easily hungry

Description:

an adorable dog that will make you happy!

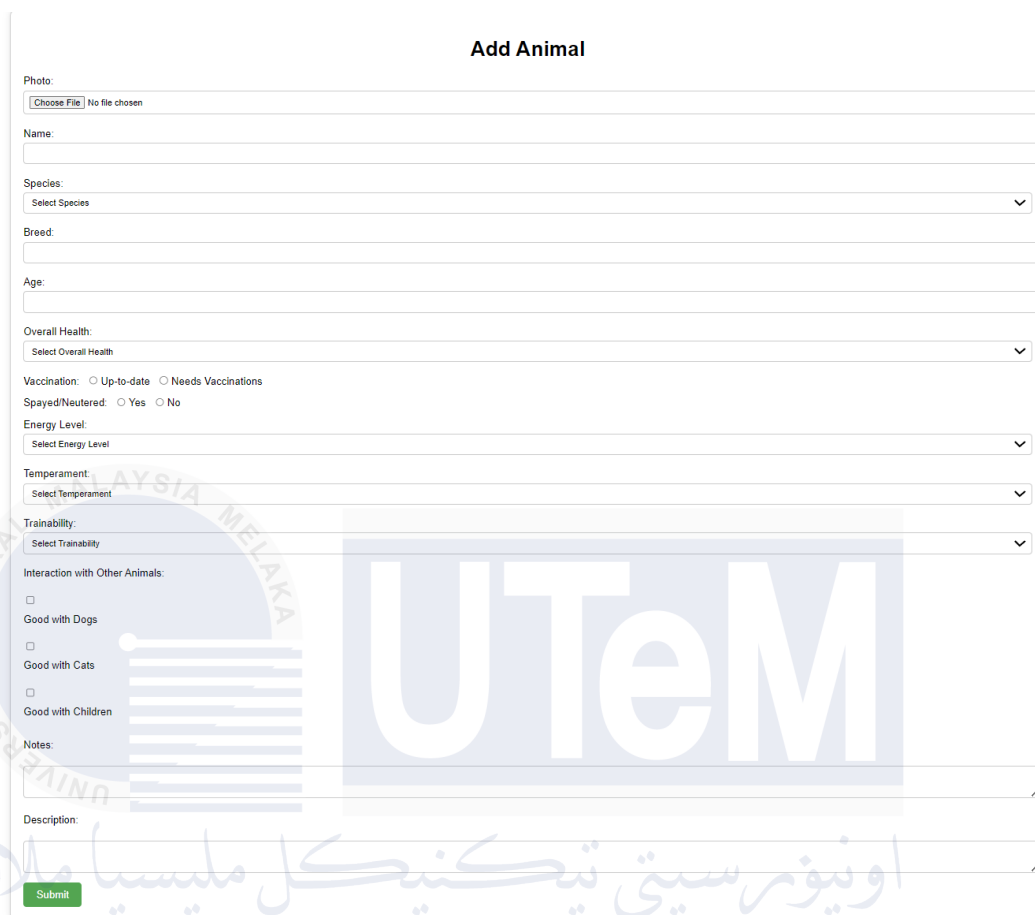
[Update Profile](#)

[Delete Profile](#)

[Medical Record](#)

[Moment Record](#)

Figure 4.4.3.6 Add Animal



Add Animal

Photo:
 No file chosen

Name:

Species:

Breed:

Age:

Overall Health:

Vaccination: ☐ Up-to-date ☐ Needs Vaccinations
 Spayed/Neutered: ☐ Yes ☐ No

Energy Level:

Temperament:

Trainability:

Interaction with Other Animals:

☐ Good with Dogs

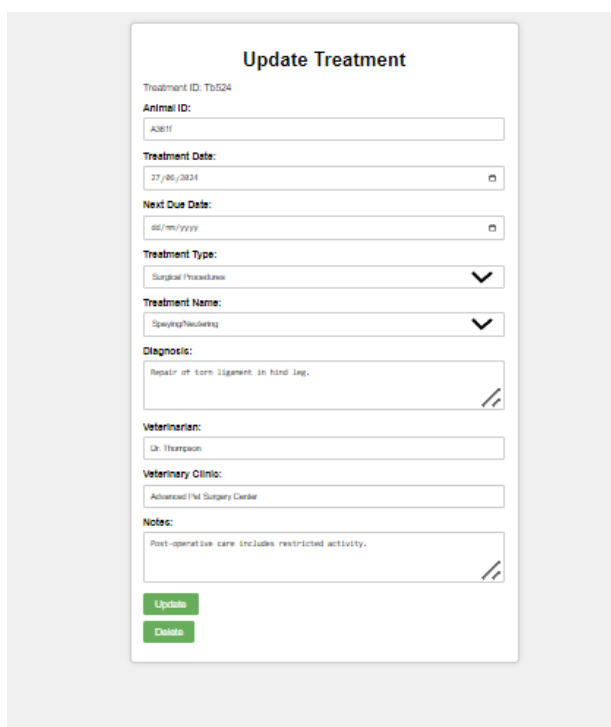
☐ Good with Cats

☐ Good with Children

Notes:

Description:

Figure 4.4.3.7 Update Treatment



Update Treatment

Treatment ID: Tb524

Animal ID:

Treatment Date:

Next Due Date:

Treatment Type:

Treatment Name:

Diagnosis:

Veterinarian:

Veterinary Clinic:

Notes:

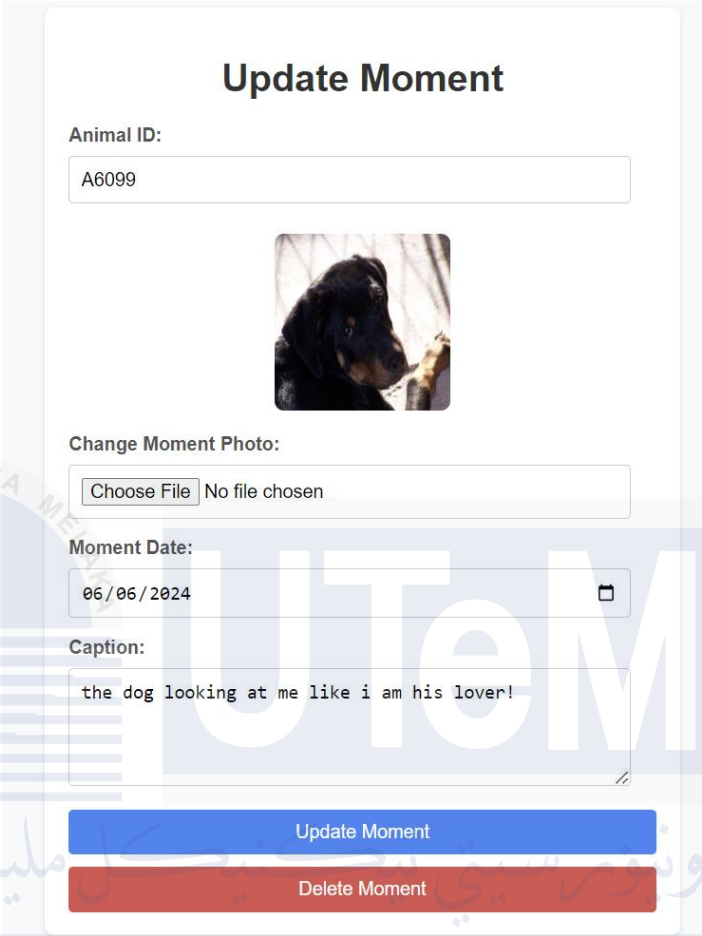
Figure 4.4.3.8 Add treatment

The image shows a web form titled "Add Treatment" with a large "UTeM" watermark in the background. The form contains the following fields:

- Animal ID:
- Treatment Date: 
- Next Due Date: 
- Treatment Type:
- Treatment Name:
- Diagnosis:
- Veterinarian:
- Veterinary Clinic:
- Notes:
-

اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.4.3.9 Update Moment



Update Moment

Animal ID: A6099



Change Moment Photo: Choose File No file chosen

Moment Date: 06/06/2024

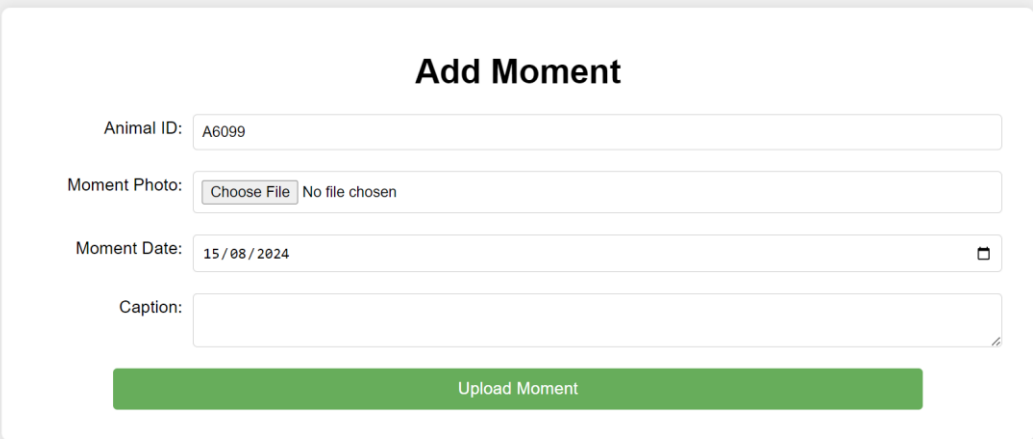
Caption: the dog looking at me like i am his lover!

Update Moment

Delete Moment

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.4.3.10 Add Moment



Add Moment

Animal ID: A6099

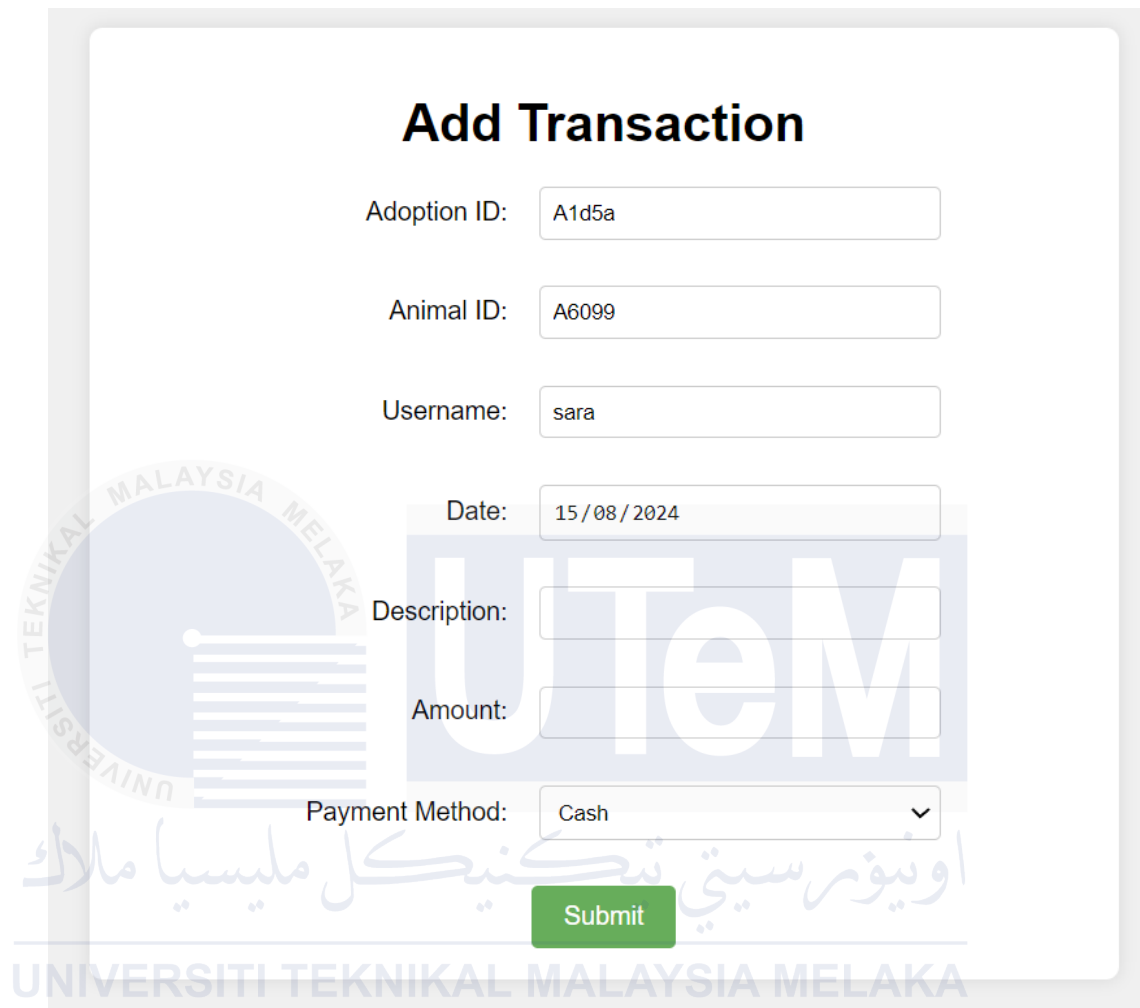
Moment Photo: Choose File No file chosen

Moment Date: 15/08/2024

Caption:

Upload Moment

Figure 4.4.3.11 Add Transaction



Add Transaction

Adoption ID:

Animal ID:

Username:

Date:

Description:

Amount:

Payment Method:

Figure 4.4.3.12 Adoption Detail Update

Search: Filter by Status:

AdoptionID	AnimalID	AnimalName	Username	PhoneNumber	EmailAddress	AdoptionStatus	MeetingLocation	MeetingNote	SubmissionDate	Action
A08e9	A6099	dog grey	a	017289212	asim2@gmail.com	Adoption Success		Date and Time Suggestions: 1. 2024-08-06T04:52 2. 3.	2024-08-05 14:01:26.086000	Delete
Ae5a9	A59c7	Cat2	a	017289212	asim2@gmail.com	Apply		Date and Time Suggestions: 1. 2024-08-08T22:54 2. 3.	2024-08-03 21:53:57.683000	Delete
Ada1a	A6099	dog grey	a	017289212	asim2@gmail.com	Meet Up		Date and Time Suggestions: 1. 2024-07-09T14:16 2. 2024-07-11T17:16 3. 2024-07-09T14:14	2024-07-16 14:14:19.862000	Delete
A07c1	A6099	dog grey	a	017289212	asim2@gmail.com	Apply		Date and Time Suggestions: 1. 2024-07-17T16:13 2. 2024-07-05T16:13 3. 2024-07-07T16:13	2024-07-16 14:12:41.056000	Delete
Adel9c	A6099	dog grey	a	017289212	asim2@gmail.com	Apply		Date and Time Suggestions: 1. 2024-07-04T16:04 2. 2024-07-16T16:04 3. 2024-07-04T17:06	2024-07-16 14:03:15.999000	Delete
A3157	A6099	dog grey	a	017289212	asim2@gmail.com	Apply		Date and Time Suggestions: 1. 2024-07-04T22:30 2. 2024-07-19T10:30 3. 2024-07-13T10:30	2024-07-15 21:29:33.744000	Delete

Figure 4.4.3.13 Browse Animals

Browse Animals

Species:

Age:

[Age ranges](#)

Search:

[Apply Filters](#) [Reset Filters](#)



dog grey

an adorable dog that will make you happy!

[Adopt Me](#)



Cat2

Siamese cats are sleek and elegant with a short coat and striking blue almond-shaped eyes. They are vocal and affectionate, often forming strong bonds with their owners.

[Adopt Me](#)



Cat3

Persian cats are characterized by their long, luxurious fur and round faces. They have a calm and sweet



Cat6

Sphynx cats are hairless with wrinkled skin and large ears. They have a warm and affectionate personality, enjoying

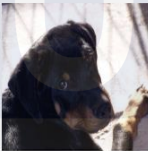
Figure 4.4.3.14 Animal Details

Animal Details



Name: dog grey
 Species: dog
 Breed: chia
 Age: 5
 Overall Health: Good
 Vaccination: up-to-date
 Spayed/Neutered: yes
 Energy Level: medium
 Temperament: anxious
 Trainability: independent
 Interaction with Other Animals: good_with_dogs, good_with_children
 Date of Intake: 2024-06-25
 Description: an adorable dog that will make you happy!

Moment Record



Moment Date: 2024-06-06
 Caption: the dog looking at me like i am his lover!

[Schedule Adoption Meeting](#)

Figure 4.4.3.15 Appointment Information

Appointment Information	
Adoption ID	A08e9
Animal ID	A6099
Animal Name	dog grey
Adoption Status	adoption_success
Meeting Location	
Submission Date and Time	2024-08-05 14:01:26.086000
Adoption ID	Ae5a9
Animal ID	A59c7
Animal Name	Cat2
Adoption Status	apply
Submission Date and Time	2024-08-03 21:53:57.683000

Figure 4.4.3.16 Latest Report

Latest Report

Search ID:

Filter Animal: All

Filter Status: All

Start Date:

End Date:

Report ID	Animal Type	Location	Date	Status	Action
R6ac3	dog	2.3080519823232692 102.31888630232108	2024-07-25 00:00:00	Chec ▼	Edit
Rc67f	dog	3.1275308788720015 101.72819660236766	2024-06-23 00:00:00	Unch ▼	Edit
R8tb8	dog	3.157838103344537 101.67991291814994	2024-06-23 00:00:00	Unch ▼	Edit
Rf3ac	dog	3.134574092618977 101.69783853653672	2024-06-23 00:00:00	Unch ▼	Edit
R529e	dog	3.1468683981961103 101.68042790233316	2024-06-23 00:00:00	Unch ▼	Edit
R5985	dog	3.155095687904553 101.70034062408538	2024-06-23 00:00:00	Unch ▼	Edit
Rc779	dog	3.1573239010009844 101.69364582970314	2024-06-23 00:00:00	Unch ▼	Edit
Rf6b6	dog	3.15800950406915 101.69124257018136	2024-06-23 00:00:00	Unch ▼	Edit
R64f2	dog	3.156981099297188 101.6830028232494	2024-06-23 00:00:00	Unch ▼	Edit

Figure 4.4.3.17 Report Detail

Report Detail



Report ID: R6ac3
Animal Type: dog
Location: 2.3080519823232692, 102.31888630232108
Description: At ftkm, i saw a grey color dog
Phone Number: 01762837628
Date: 2024-07-25 00:00:00
Time: 12:30
Status: Checked ▼
Remarks:

Figure 4.4.3.18 View Animal

View Animal									
Filter by Species Filter Health Filter Vaccination Filter Spayed/Neutered Filter Date Clear Add Filter									
Animal ID	Name	Species	Breed	Age	Overall Health	Vaccination	Spayed/Neutered	Date of Intake	Action
A0099	dog grey	dog	chia	5	Good	up-to-date	yes	2024-06-25	Edit
A59c7	Cat2	cat	Siamese	2	Good	needs-vaccinations	yes	2024-06-23	Edit
A8084	Cat3	cat	Persian	5	Good	up-to-date	no	2024-06-23	Edit
A382	Cat4	cat	Bengal	1	Fair	up-to-date	no	2024-06-23	Edit
A6c3f	Cat5	cat	Ragdoll	4	Fair	needs-vaccinations	no	2024-06-23	Edit
A2686	Cat6	cat	Sphynx	2	Excellent	up-to-date	yes	2024-06-23	Edit
A361f	Cat7	cat	British Shorthair	6	Good	needs-vaccinations	yes	2024-06-23	Edit
A2940	Cat8	cat	Russian Blue	3	Critical	needs-vaccinations	no	2024-06-23	Edit
A64c3	Cat9	cat	Scottish Fold	2	Excellent	up-to-date	yes	2024-06-23	Edit
A19a2	Cat10	cat	Burmese	4	Good	needs-vaccinations	no	2024-06-23	Edit
A34e0	Dog1	dog	Labrador Retriever	2	Good	up-to-date	yes	2024-06-23	Edit

Figure 4.4.3.19 Animal Profile Detail



Animal Profile Update



Animal ID: A0099

New Photo: [Choose File](#) No file chosen

Name: dog grey

Species: Dog

Breed: chia

Age: 5

Overall Health: Good

Vaccination: ☒ Up-to-date ☐ Needs Vaccinations

Spayed/Neutered: ☒ Yes ☐ No

Energy Level: Medium

Temperament: Anxious

Trainability: Independent

Interaction with Other Animals: ☒ Good with Dogs ☐ Good with Cats ☒ Good with Children

Date of Intake: 25/06/2024

Last Grooming Date: 11/06/2024

Notes: he is easily hungry

Description: an adorable dog that will make you happy!

[Update Profile](#)

[Delete Profile](#)

[Medical Record](#)

[Movement Record](#)

Figure 4.4.3.20 Medical Record

Medical Record
Animal ID: A6099

Add Treatment

Treatment ID	Animal ID	Treatment Date	Next Due Date	Treatment Type	Treatment Name	Actions
T7f0f	A6099	2024-06-25		Preventative Care	Deworming	Edit

Figure 4.4.3.21 Treatment Detail

Update Treatment
 Treatment ID: T7f0f
 Animal ID:
 Treatment Date:
 Next Due Date:
 Treatment Type:
 Treatment Name:
 Diagnosis:
 Veterinarian:
 Veterinary Clinic:
 Notes:

[Update](#)
[Delete](#)

Figure 4.4.3.22 Moment Record

Moment Record

Animal ID: A6099

Add Moment



Moment Date: 2024-06-06
Caption: the dog looking at me like i am his lover!

Edit Post

Figure 4.4.3.23 Appointment List

Search:

Filter by Status:

All

Done

AdoptionID	AnimalID	AnimalName	Username	PhoneNumber	EmailAddress	AdoptionStatus	MeetingLocation	MeetingNote	SubmissionDate	Action
A06a9	A6099	dog grey	a	017209212	asin2@gmail.com	Adoption Success		Date and Time Suggestions: 1, 2024-08-08T04:52 Z, 2	2024-08-05 14:01:26.080000	Cancel
A06a9	A6097	Cat2	a	017209212	asin2@gmail.com	Pending		Date and Time Suggestions: 1, 2024-08-08T22:54 Z, 2	2024-08-03 21:53:57.000000	Cancel
A06fa	A6099	dog grey	a	017209212	asin2@gmail.com	Pending		Date and Time Suggestions: 1, 2024-07-09T14:16 Z, 2024-07-11T17:16 Z, 2024-07-09T14:14	2024-07-16 14:14:19.000000	Cancel
A07c1	A6099	dog grey	a	017209212	asin2@gmail.com	Pending		Date and Time Suggestions: 1, 2024-07-17T16:13 Z, 2024-07-08T16:13 Z, 2024-07-07T16:13	2024-07-16 14:12:41.000000	Cancel
A06fc	A6099	dog grey	a	017209212	asin2@gmail.com	Pending		Date and Time Suggestions: 1, 2024-07-04T16:04 Z, 2024-07-16T16:04 Z, 2024-07-04T17:06	2024-07-16 14:03:15.000000	Cancel
A3107	A6099	dog grey	a	017209212	asin2@gmail.com	Pending		Date and Time Suggestions: 1, 2024-07-04T22:30 Z, 2024-07-19T19:30 Z, 2024-07-13T10:30	2024-07-16 21:29:33.740000	Cancel
A0706	A6099	dog grey	a	017209212	asin2@gmail.com	Pending			2024-06-30 00:23:01.400000	Cancel
A106a	A6099	dog grey	sara	016074603	sara@gmail.com	Adoption Success	FTMK lobby	10pm please wear tie	2024-06-25 12:44:36.000000	Cancel
A0916	A106a	Dog3	niko	017203602	niko@gmail.com	Pending			2024-06-23 20:36:23.400000	Cancel
A0604	A0603	Dog1	niko	017203602	niko@gmail.com	Pending			2024-06-23 20:36:18.200000	Cancel

Figure 4.4.3.24 Analysis

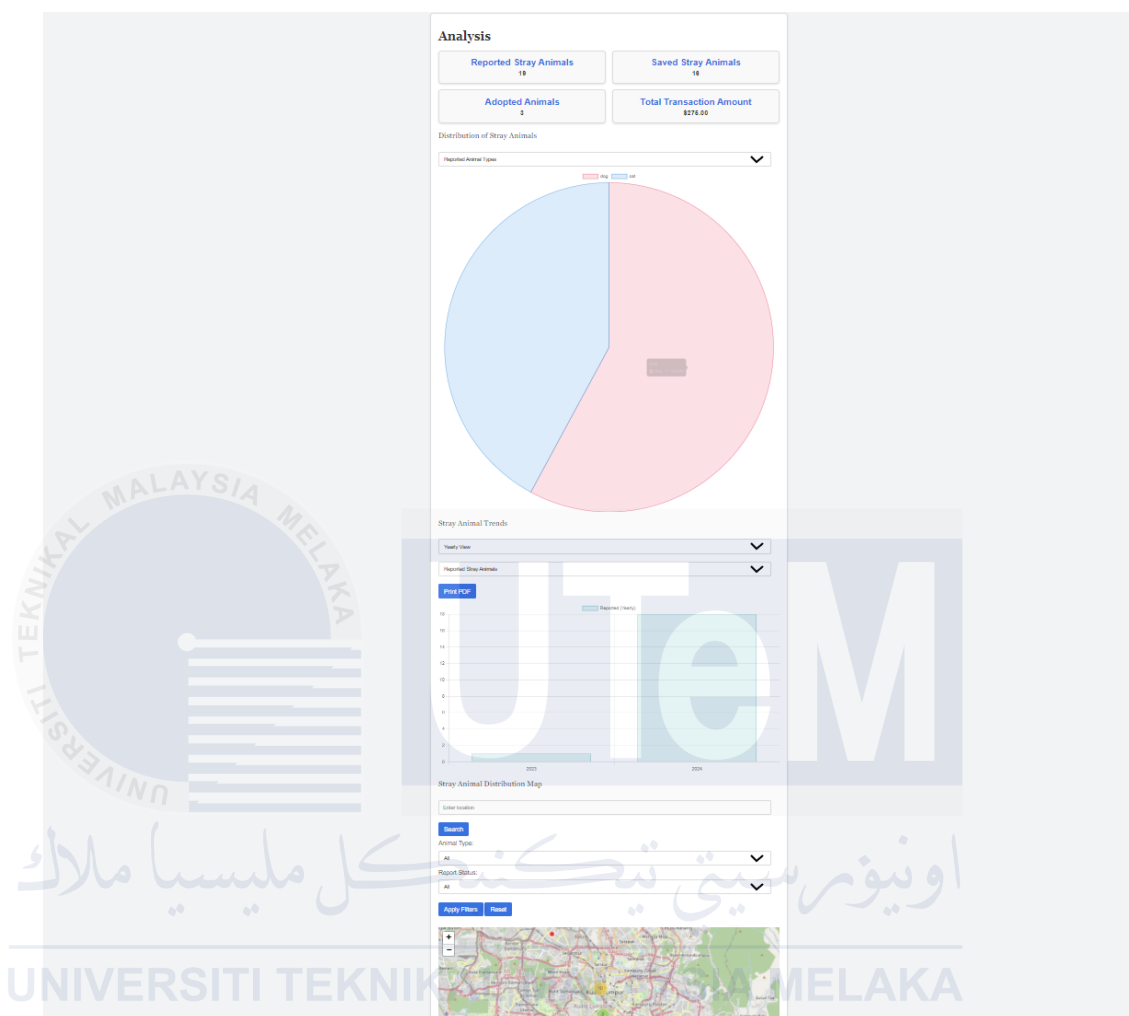


Figure 4.4.3.25 Analysis Dashboard

Analysis

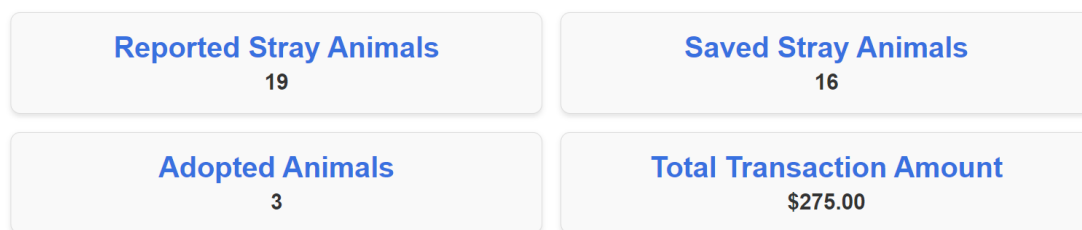


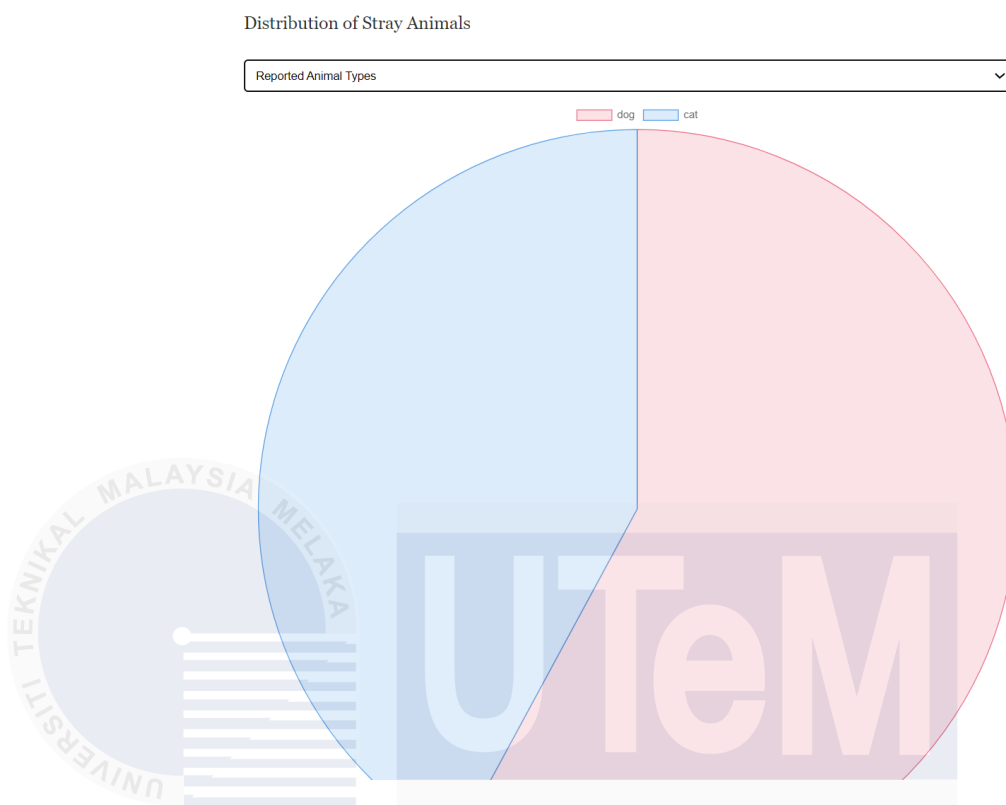
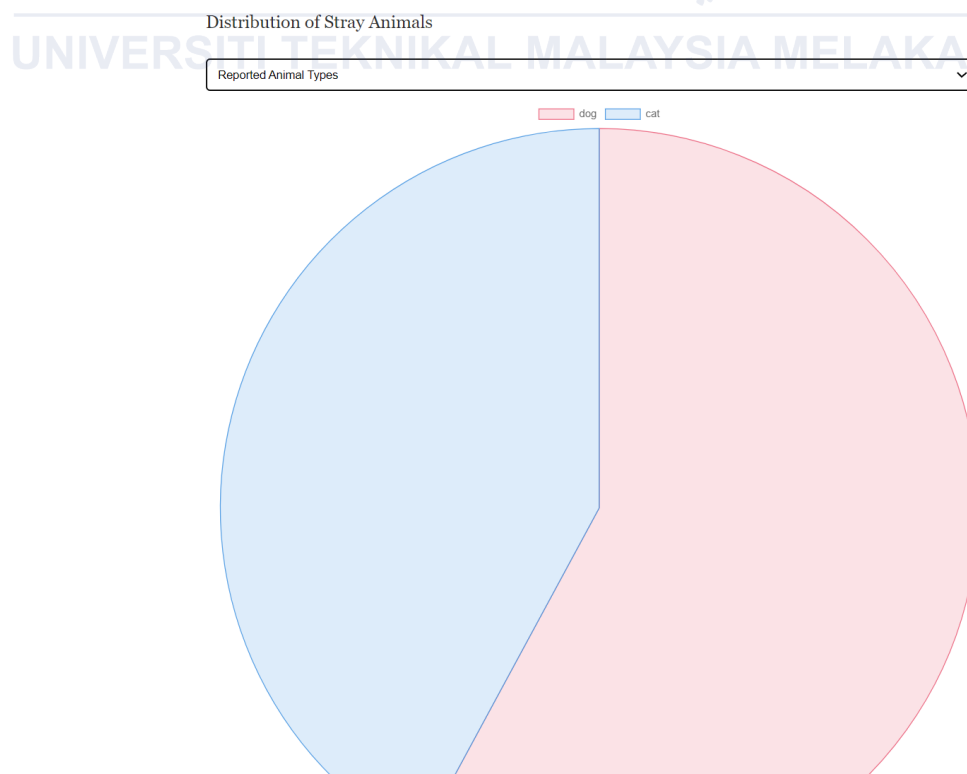
Figure 4.4.3.26 Pie chart filter by reported stray Animal types**Figure 4.4.3.27 Pie chart filter by reported stray Animal types**

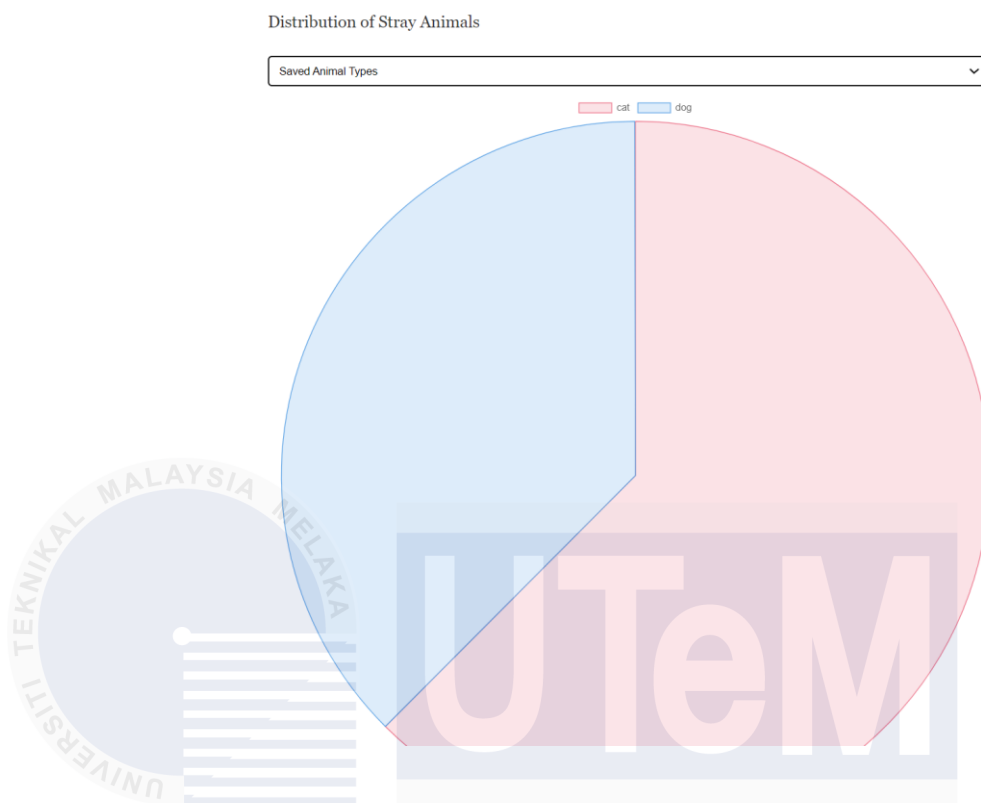
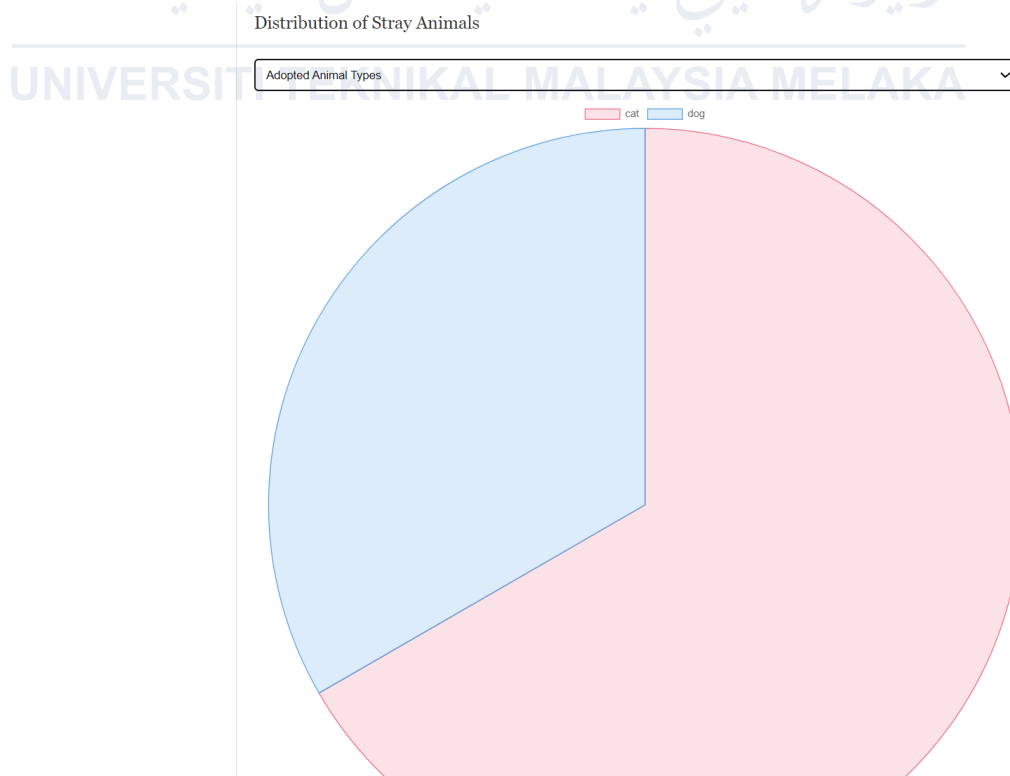
Figure 4.4.3.28 Pie chart filter by saved animal types**Figure 4.4.3.29 Pie chart filter by adopted animal types**

Figure 4.4.3.30 Stray Animal Trends Yearly View Chart

Stray Animal Trends

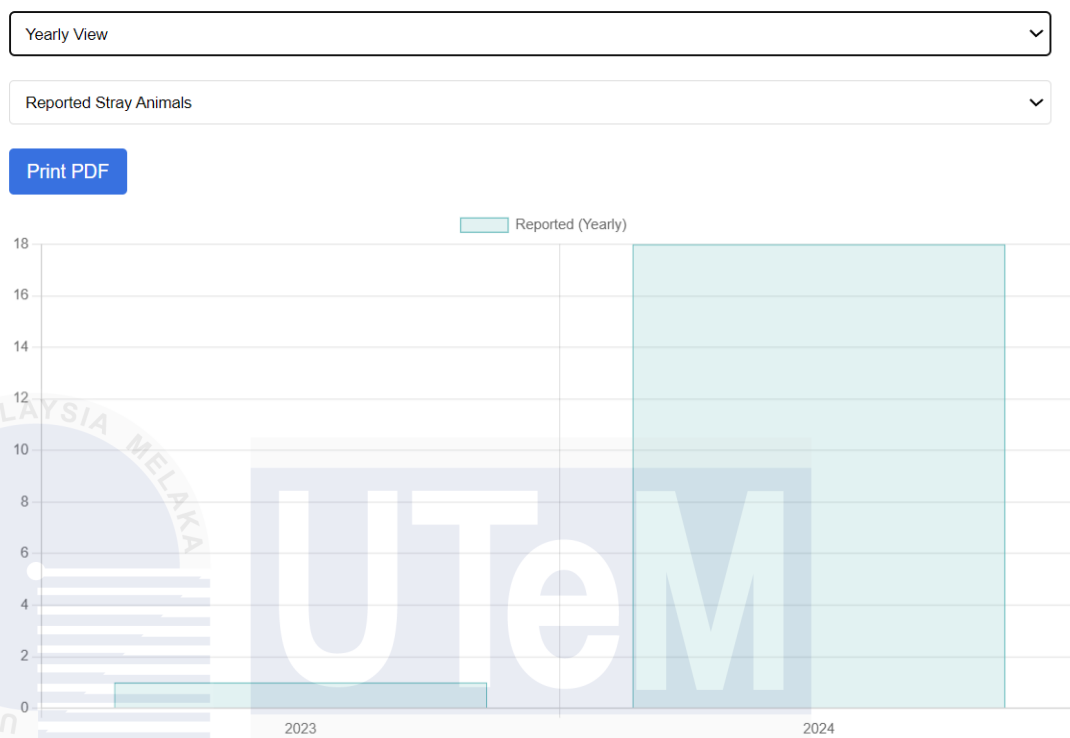


Figure 4.4.3.31 Stray Animal Trends Monthly View Selection

Stray Animal Trends

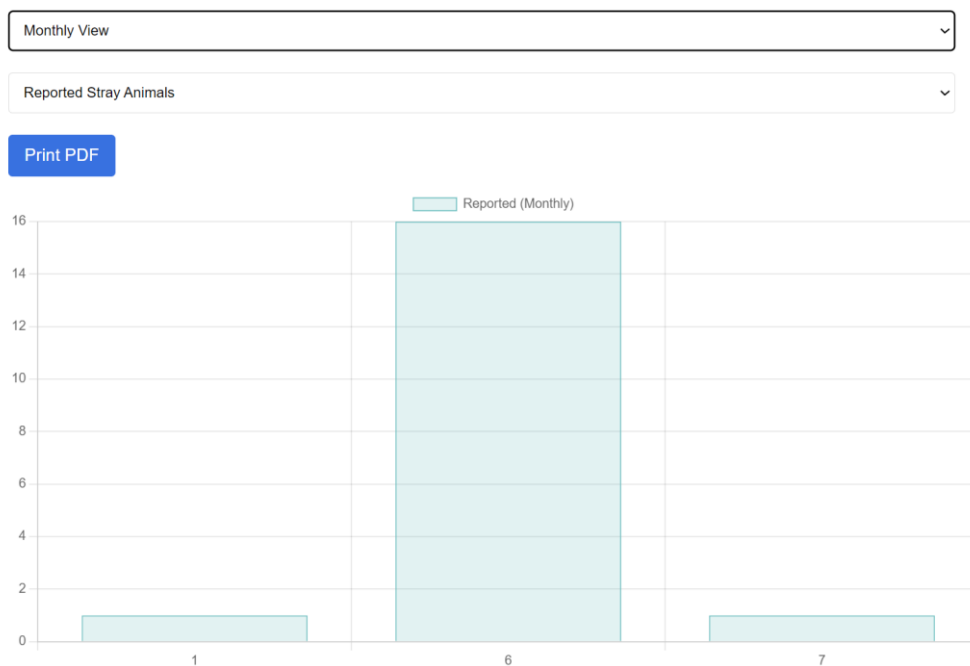


Figure 4.4.3.32 Adopted Animal Chart

Stray Animal Trends

Yearly View

Adopted Animals

Print PDF

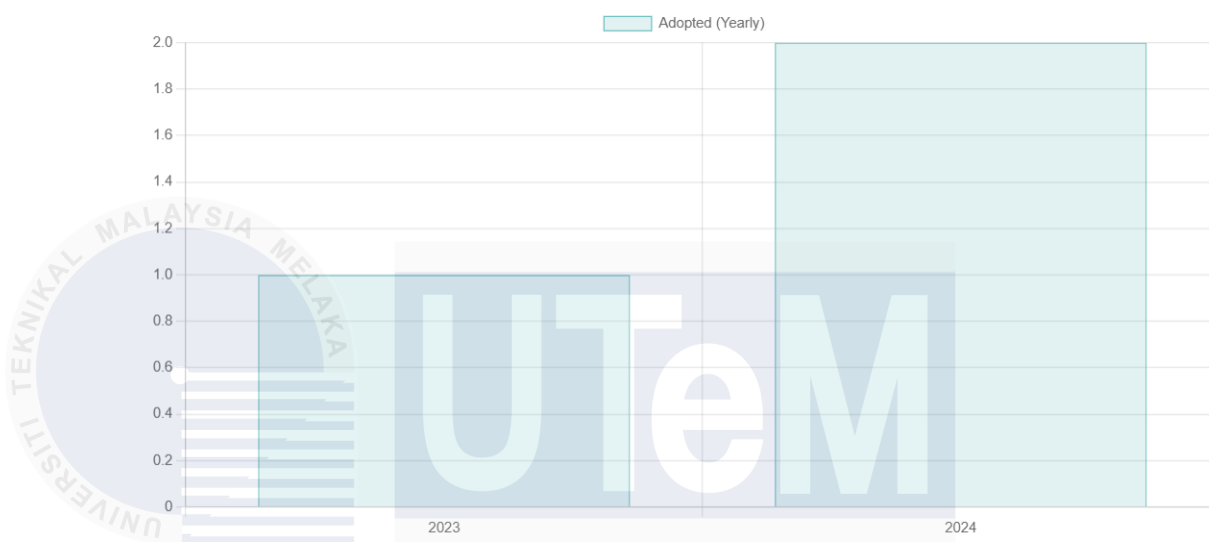


Figure 4.4.3.33 Transaction Amount Chart

Stray Animal Trends

Yearly View

Transaction Amount

Print PDF

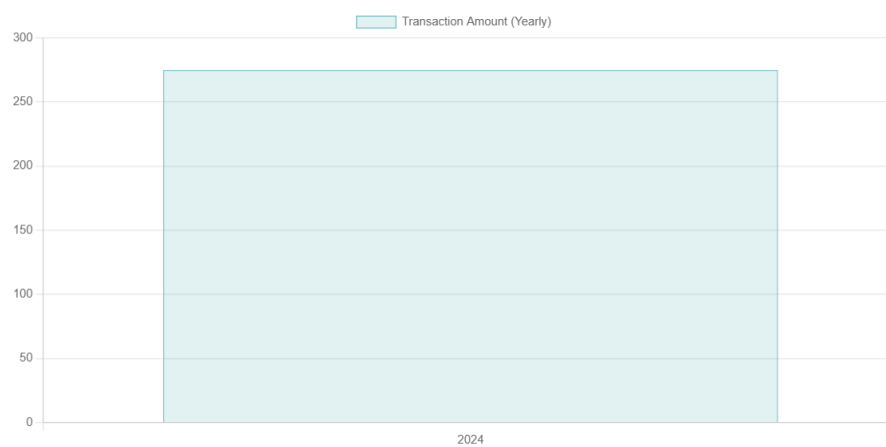


Figure 4.4.3.34 Exported Stray Animal Trends PDF

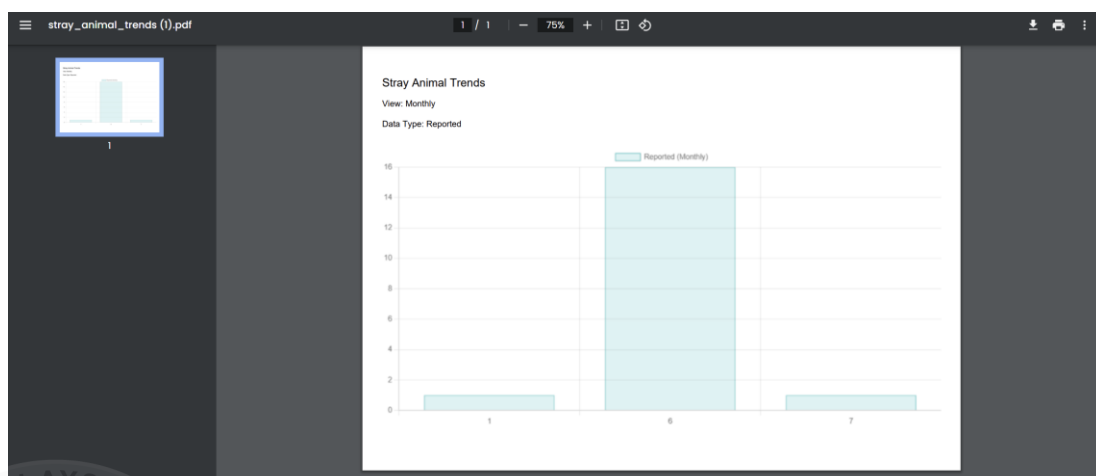


Figure 4.4.3.35 Stray Animal Distribution Map Location Search

Stray Animal Distribution Map

Animal Type:

Report Status:

Leaflet | © OpenStreetMap contributors

Figure 4.4.3.36 Stray Animal Distribution for Dog

Stray Animal Distribution Map

Animal Type:

Dog
 ▼

Report Status:

All
 ▼

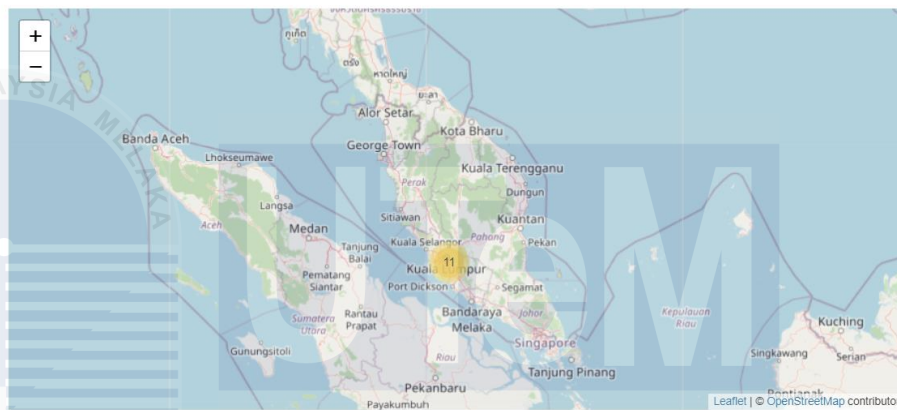


Figure 4.4.3.37 Stray Animal Distribution for Cat

Stray Animal Distribution Map

Animal Type:

Cat
 ▼

Report Status:

All
 ▼

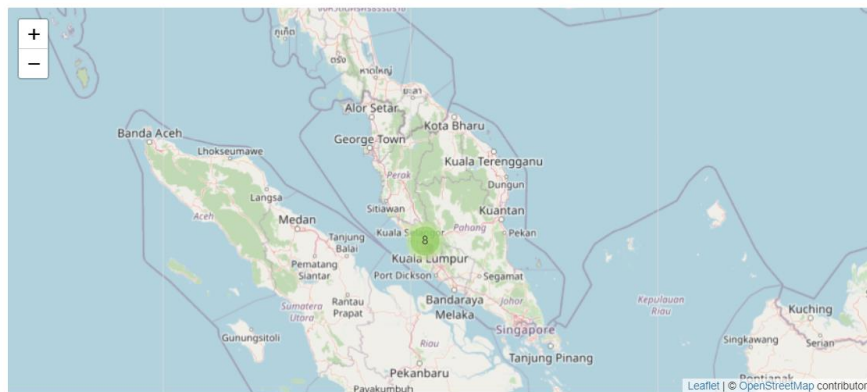


Figure 4.4.3.38 Stray Animal Distribution Map Report Status Checked

Stray Animal Distribution Map

Enter location

Search

Animal Type:

All

Report Status:

Checked

Apply Filters **Reset**

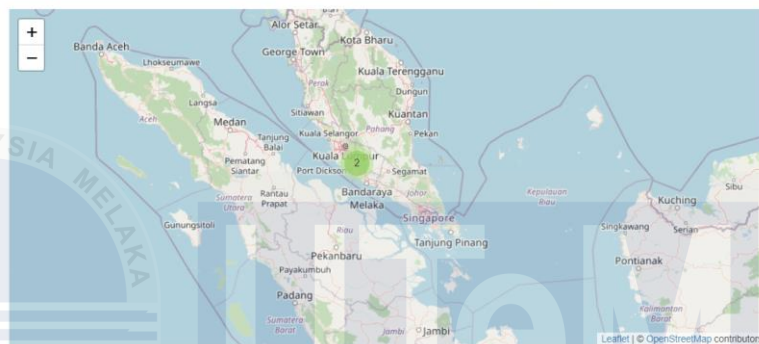


Figure 4.4.3.39 Stray Animal Distribution Map Report Status Unchecked

Stray Animal Distribution Map

Enter location

Search

Animal Type:

All

Report Status:

Unchecked

Apply Filters **Reset**

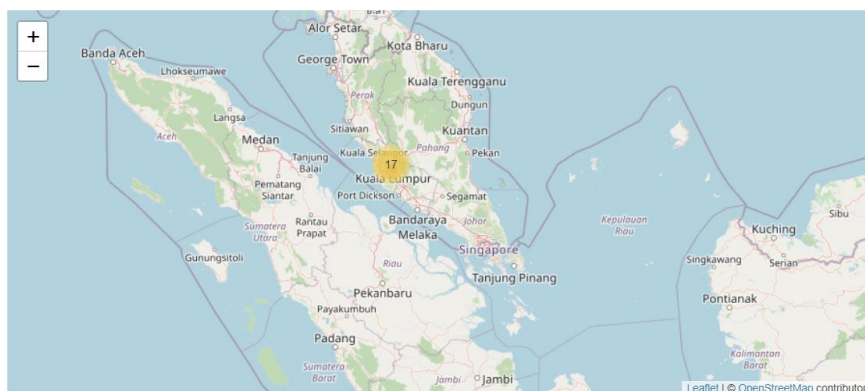
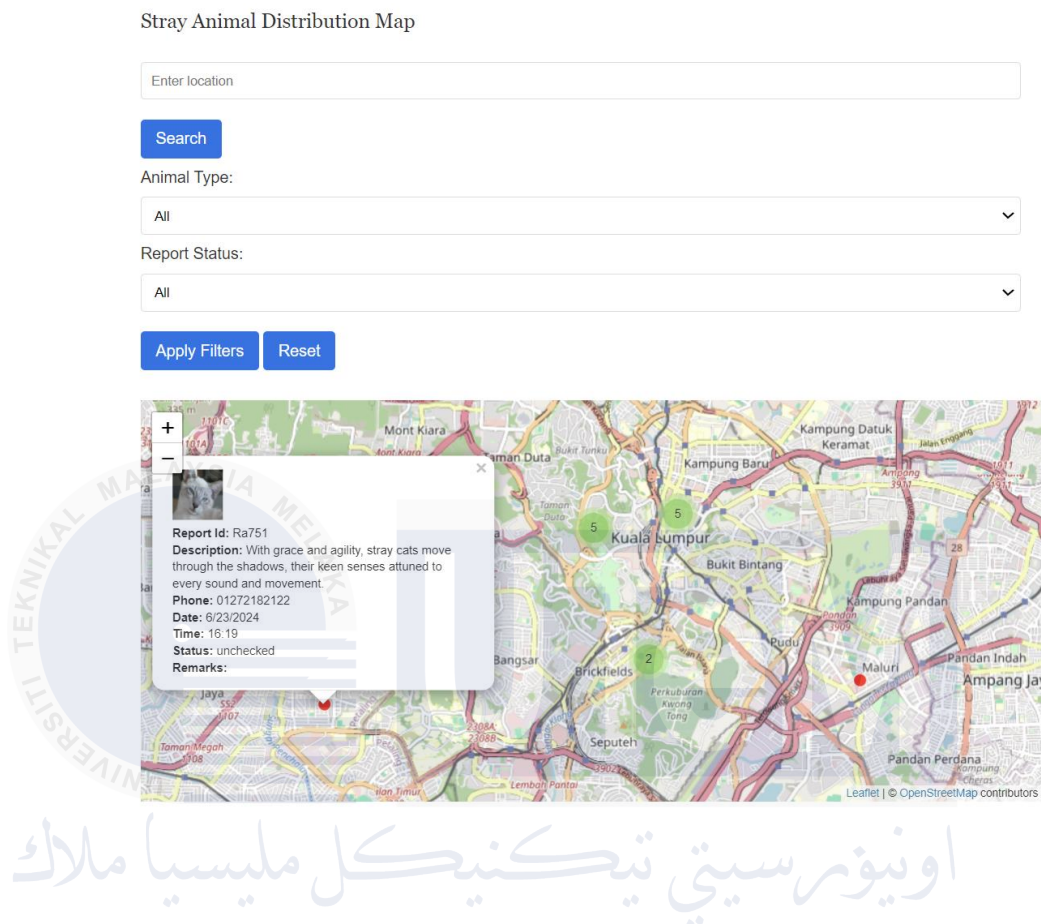


Figure 4.4.3.40 Stray Animal Distribution Map pop up from map points



4.5 Conclusion

This chapter concludes the design phase of the Find Your Pawfect Buddy system, showcasing a systematic approach to transforming the requirements and features identified in the analysis phase into a comprehensive design solution. By addressing key aspects such as database structure, user interface, system architecture, security measures, and integration planning, we have created a robust blueprint for the animal adoption platform. This design phase marks the crucial transition from problem domain to solution domain, providing a clear roadmap for the subsequent implementation, testing, and maintenance phases. The resulting design document will guide developers in creating a system that effectively connects animals with potential adopters, streamlines shelter operations, and ensures a positive user experience. By thoroughly addressing these design considerations, we have laid a strong foundation for a system that will make a meaningful impact in the world of animal adoption, ultimately helping more animals find their forever homes.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

In this chapter, the task of implementing the database design for the "Find Your Pawfect Buddy" system is presented. The database installation and configuration procedures are detailed. During the database implementation phase, MongoDB is installed on the Windows 11 platform. In this phase, both Data Definition Language (DDL) and Data Manipulation Language (DML) in the form of MongoDB queries are implemented. The implementation status for each module is described.

5.2 System Development Environment Setup

For the "Find Your Pawfect Buddy" system, the development environment consists of HTML, CSS, and JavaScript for the frontend, Python with Flask for the backend, and MongoDB for the database, chosen for their robustness, scalability, and ease of use. The primary tools used include Visual Studio Code for coding and MongoDB Compass for database management. On the Windows 11 platform, MongoDB can be downloaded and installed from its official website, with setup involving configuring MongoDB as a service to start automatically. These components are essential for the development and deployment of the application.

5.2.1 Steps of Installation Setup

Step 1: To get the MongoDB Community Server shown in figure 5.2.1.1, visit the MongoDB get Center.

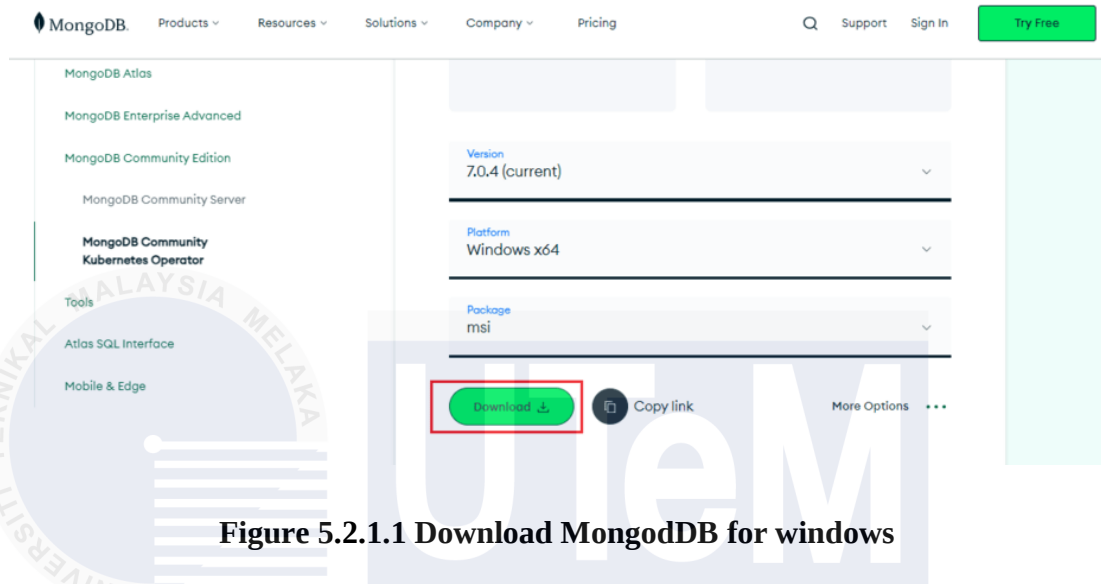


Figure 5.2.1.1 Download MongoDB for windows

Step 2: Following the download, launch the Microsoft Installer file and select the next button from the startup page (see figure 5.2.1.2).



Figure 5.2.1.2 Startup Screen

Step 3: Now click the next button in figure 5.2.1.3 after accepting the End-User License Agreement:

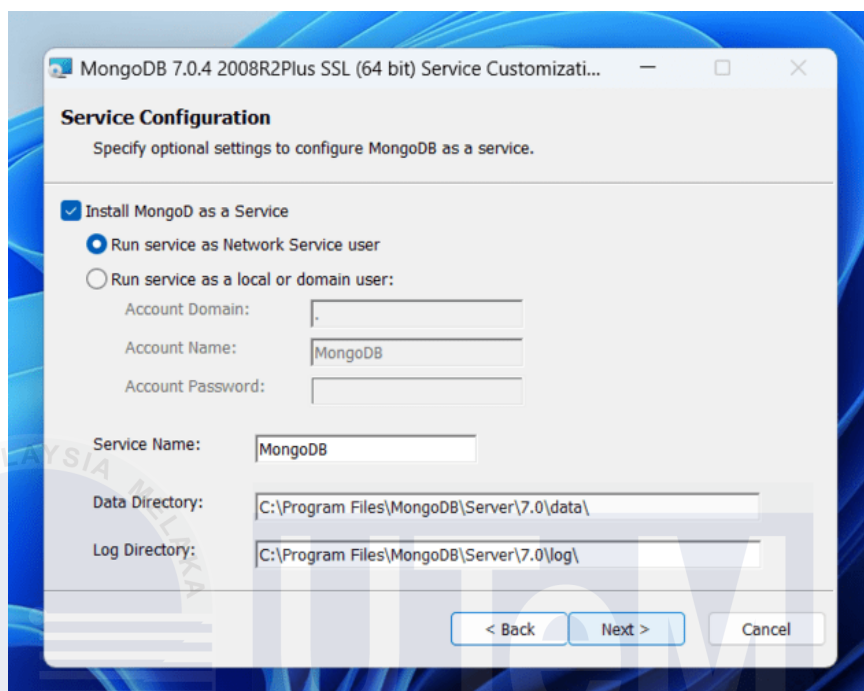


Figure 5.2.1.3 Service Configuration

Step 4: Choose the full option now to install every functionality of the application. Here, utilize the Custom option shown in figure 5.2.1.4 if you would like to install only specific program features and choose where the installation will be made.

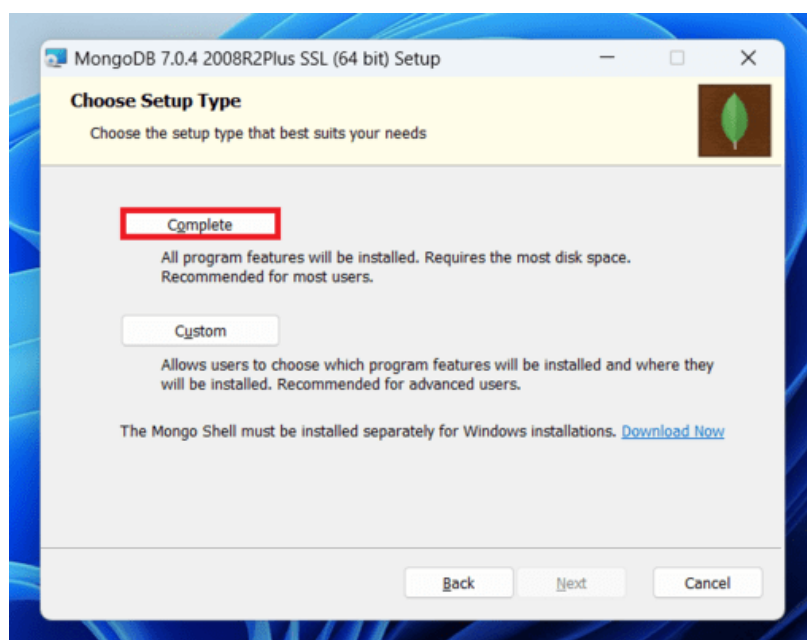


Figure 5.2.1.4 Choose Setup Type

Step 5: After choosing to "Run service as Network Service user," copy the data directory's path. In figure 5.2.1.5, click Next.

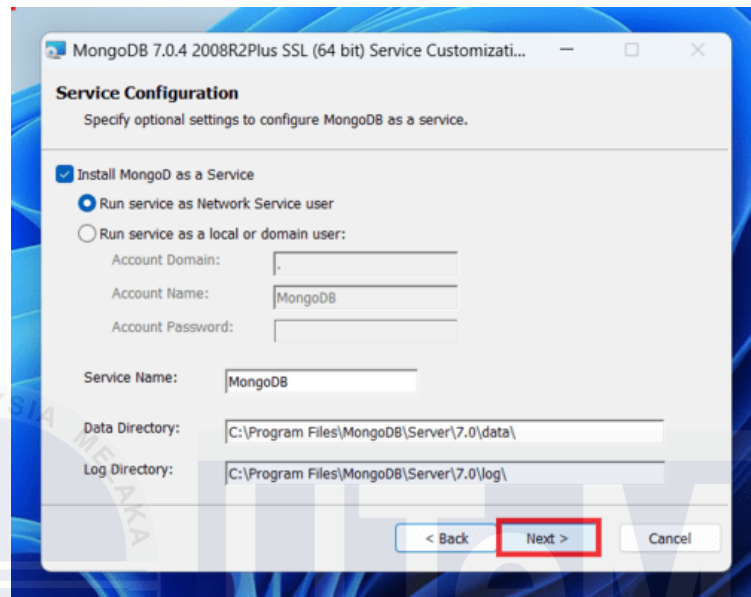


Figure 5.2.1.5 Service Configuration

Step 6: To begin the MongoDB installation process, click the Install button in figure 5.2.1.6.

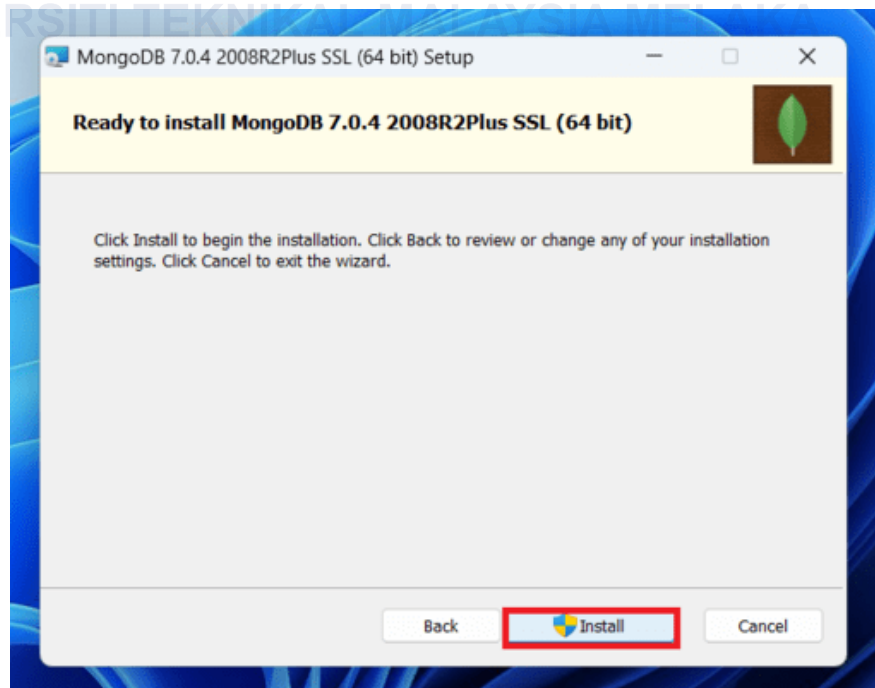


Figure 5.2.1.6 Install MongoDB

Step 7: Figure 5.2.1.7 shows the installation of MongoDB after selecting the install button.

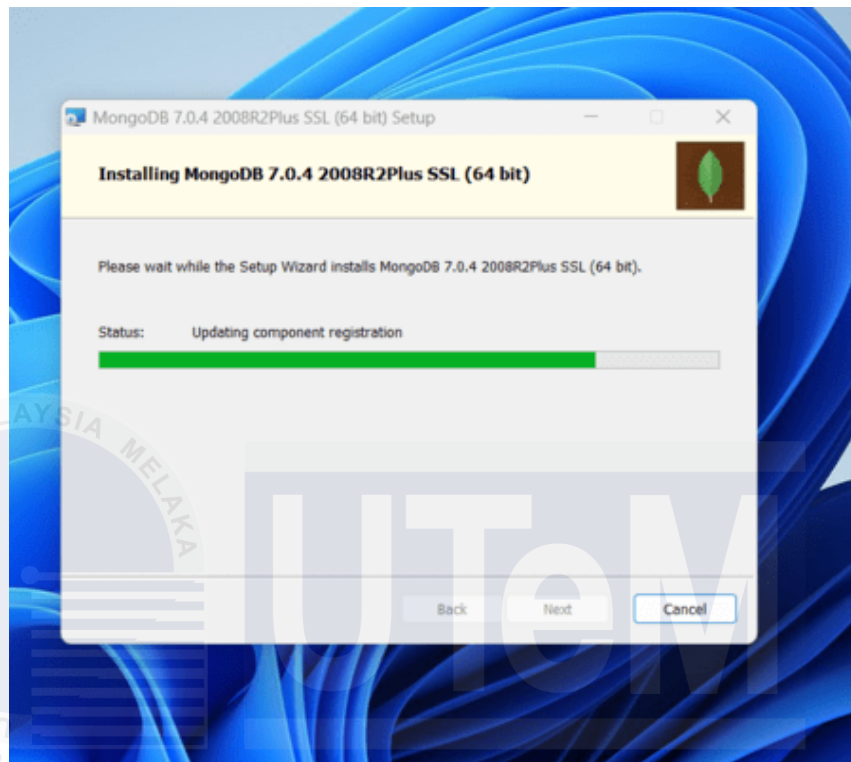


Figure 5.2.1.7 Installing Process

Step 8: To finish the MongoDB installation process, click the Finish button now.

5.3 Database Implementation

5.3.1 Create Table Commands

Below shows the command to create the collection in MongoDB:

```
// Create collections
```

```
db.createCollection("adoption");
```

```
db.createCollection("animal");
```

```
db.createCollection("medical");
```

```
db.createCollection("moment");
```

```
db.createCollection("report");
```

```
db.createCollection("staff");
```

```
db.createCollection("transaction");
```

```
db.createCollection("user");
```

5.3.2 Insert Data into Collections

1. Adoption Collection

```
db.adoption.insertOne({  
  _id: ObjectId("666121049dcbb40c7d168733"),
```

```
  adoption_id: "Adopt1234",
```

```
  animal_id: "Animal5678",
```

```
  animal_name: "Buddy",
```

```
  username: "john_doe",
```

```
  phone_number: "3125556789",
```

```
  email_address: "johndoe@example.com",
```

```
adoption_status: "applied",

meeting_location: "Shelter Main Office",

meeting_note: "Initial meeting scheduled for Monday",

submission_datetime: ISODate("2024-06-06T10:37:56.883Z")

});
```

2. Animal Collection

```
db.animal.insertOne({
  _id: ObjectId("661fb10dafa6ec8a6870f828"),
```

```
  animal_id: "Animal5678",
```

```
  species: "dog",
```

```
  breed: "Labrador Retriever",
```

```
  age: 4,
```

```
  overall_health: "Excellent",
```

```
  vaccination: "up-to-date",
```

```
  spayed_neutered: "yes",
```

```
  energy_level: "high",
```

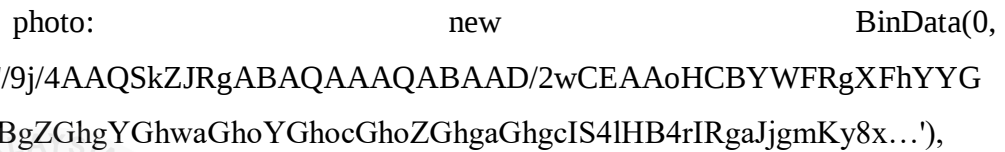
```
  temperament: "friendly",
```

trainability: "high",

interaction_with_other_animals: ["good_with_cats", "good_with_dogs"],

date_of_intake: "2024-04-17",

notes: "Loves to play fetch",

photo:  new BinData(0, '9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAoHCBYWFRgXFhYYG BgZGhgYGhwaGhoYGhocGhoZGhgaGhgclS4lHB4rIRgaJjgmKy8x...'),

last_grooming_date: "2024-04-24",

description: "Buddy is a playful and energetic Labrador looking for a loving home.",

name: "Buddy"

});

3. Medical Collection

db.medical.insertOne({

_id: ObjectId("662718275fbb4112cfa0cb2b"),

treatment_id: "Treat001",

animal_id: "Animal5678",

treatment_date: "2024-04-02",

next_due_date: "2024-10-02",

diagnosis: "Allergies",

treatment_type: "Medication",

treatment_name: "Antihistamine",

medication_name: "Benadryl",

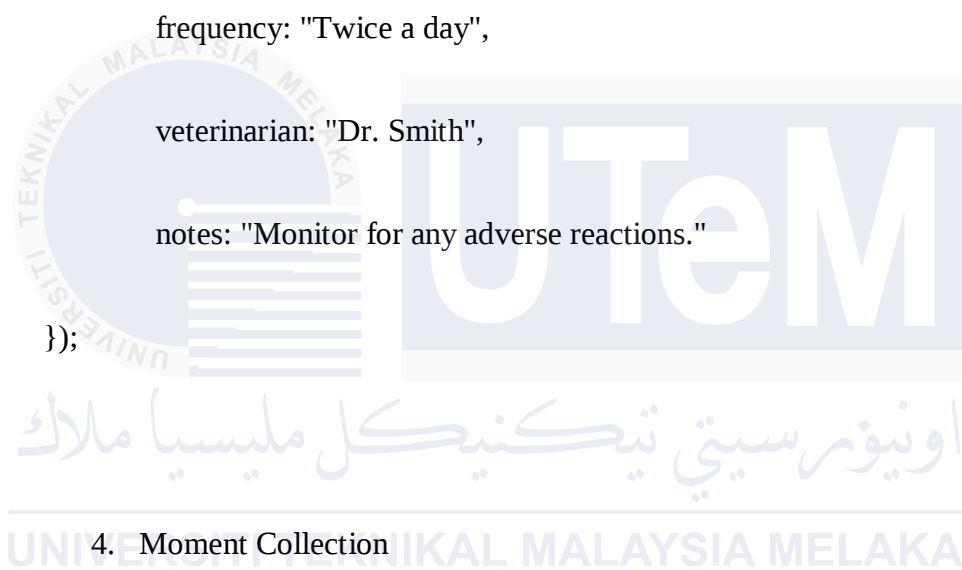
dosage: "25 mg",

frequency: "Twice a day",

veterinarian: "Dr. Smith",

notes: "Monitor for any adverse reactions."

});



4. Moment Collection

```
db.moment.insertOne({
```

```
  _id: ObjectId("662718275fbb4112cfa0cb2b"),
```

```
  moment_id: "Moment001",
```

```
  animal_id: "Animal5678",
```

```
  date: "2024-06-10",
```

```
  description: "Buddy's first beach day",
```

```

photo:                                new                                BinData(0,
'iVBORw0KGgoAAAANSUhEUgAAAYAAAAD7CAIAAADsN9ReAAIG
8klEQVR4ATxTB5IjOQxjUjvspMt3/3/f5nVot0geUD01aJcCRUCg...'),

```

```

notes: "Buddy had a great time playing in the sand and water."

```

```

});

```

5. Report Collection

```

db.report.insertOne({
  _id: ObjectId("665f09ce56cd299168602cef"),
  report_id: "Report5678",

```

```

  animal_type: "dog",

```

```

  location: "3.096550815169595, 101.73823096581337",

```

```

  description: "A large friendly bulldog seen near the park. Looks lost but
is very gentle.",

```

```

  phone_number: "8125551234",

```

```

  date: "2024-06-04",

```

```

  time: "20:33",

```

```

photo:                                new                                BinData(0,
'iVBORw0KGgoAAAANSUhEUgAAAYAAAAD7CAIAAADsN9ReA
AIG8klEQVR4ATxTB5IjOQxjUjvspMt3/3/f5nVot0geUD01aJcCRUCg
...'),

```

```

    report_status: "unchecked",

    remarks: "Follow up needed to check if the owner can be located."

});

```

6. Staff Collection



```

db.staff.insertOne({
    _id: ObjectId("666f04b53fd238fbd9ac97c5"),
    staff_id: "Staff001",
    password:
    "scrypt:32768:8:1$LxS9eGQTZwNa2uds$d1f44df310f04e9a2363e4c5816fd
    7961261...",
    phone: "0172894728",

    email: "staff@example.com",

    address: "123 Shelter St, Cityville",

    birthdate: "1985-07-15"

});

```

7. Transaction Collection

```

db.transaction.insertOne({

```

```
_id: ObjectId("666f0f39ddb8f1c29050fe9e"),
```

```
transaction_id: "Trans1234",
```

```
adoption_id: "Adopt1234",
```

```
animal_id: "Animal5678",
```

```
username: "john_doe",
```

```
date: "2024-06-19",
```

```
description: "Adoption fee",
```

```
amount: 100.00,
```

```
payment_method: "credit_card"
```



8. User Collection

```
db.user.insertOne({
```

```
_id: ObjectId("666f0f39ddb8f1c29050fe9e"),
```

```
user_id: "User1234",
```

```
username: "john_doe",
```

```
password: "hashed_password",
```

```
email: "johndoe@example.com",
```

phone_number: "3125556789",

address: "456 User Ln, Townsville",

registration_date: "2023-11-01"

});

5.3.3 Complex Queries

1. To filter animals from the animal_collection excluding those with fee_paid status in the linked 'adoption' collection, as well as including animals with missing adoption status, this pipeline uses aggregation with \$lookup, \$addFields, and \$match stages. It then sorts the results by the most recent intake date. Figure 5.3.3.1. shows the exclude fee paid query.

```

@app.route('/browse_animals')
def browse_animals():
    user_logged_in = False # Initialize user_logged_in variable
    # Check if the user is logged in
    if 'username' in session:
        user_logged_in = True

    # Fetch animals based on filter criteria
    query = {'overall_health': {'$in': ['Good', 'Excellent']}}
    species = request.args.get('species')
    age_category = request.args.get('age')
    search = request.args.get('search')
    if species:
        query['species'] = species
    if age_category:
        min_age, max_age = AGE_RANGES.get(age_category, (0, float('inf')))
        # Calculate birthdate range based on age range
        max_birth_date = datetime.now() - timedelta(days=min_age*365)
        min_birth_date = datetime.now() - timedelta(days=max_age*365)
        query['age'] = {'$gte': min_age, '$lte': max_age}

    if search:
        query['$or'] = [{'name': {'$regex': search, '$options': 'i'}}, {'description': {'$regex': search, '$options': 'i'}}]

    # Use aggregation to filter out animals with 'fee paid' status in adoption collection
    pipeline = [
        {
            '$match': query
        },
        {
            '$lookup': {
                'from': 'adoption',
                'localField': 'animal_id',
                'foreignField': 'animal_id',
                'as': 'adoption_info'
            }
        },
        {
            '$addFields': {
                'adoption_status': {
                    '$arrayElemAt': ['$adoption_info.adoption_status', 0]
                }
            }
        },
        {
            '$match': {
                '$or': [
                    {'adoption_status': {'$ne': 'fee_paid'}},
                    {'adoption_status': {'$exists': False}}
                ]
            }
        },
        {
            '$sort': {'date_of_intake': -1}
        }
    ]

    filtered_animals = list(animal_collection.aggregate(pipeline))
    return render_template('browse_animals.html', user_logged_in=user_logged_in, animals=filtered_animals)

```

Figure 5.3.3.1 Exclude Feed Paid Query

2. The code performs complex queries and aggregations to analyze various aspects of stray animals and adoption transactions. It counts the total reported stray animals, saved stray animals, and adopted pets where the adoption status is 'fee_paid.' For transactions, it aggregates and sums the transaction amounts. The code then groups the reported, saved, and adopted animals by year and month, allowing for analysis of trends over time. Additionally, it determines the distribution of animal types for reported, adopted, and saved animals. For adopted animals, it uses a lookup to join animal data from another collection based on animal_id, grouping the results by species. The purpose of these queries is to provide detailed insights into animal reports, adoptions, and financial transactions, with the data presented in the analysis dashboard. Figure 5.3.3.2 shows the analysis complex queries.

```

@app.route('/analysis')
def analysis():
    # Calculate total counts
    reported_stray_count = report_collection.count_documents({})
    saved_stray_count = animal_collection.count_documents({})
    adopted_count = adoption_collection.count_documents({'adoption_status': 'fee_paid'})

    # Aggregate the sum of 'amount' field in 'transaction' collection
    transaction_sum_pipeline = [
        {'$addFields': {'amount': {'$toDouble': '$amount'}}},
        {'$group': {'_id': None, 'total': {'$sum': '$amount'}}}
    ]
    transaction_sum_result = list(transaction_collection.aggregate(transaction_sum_pipeline))
    transaction_sum = transaction_sum_result[0]['total'] if transaction_sum_result else 0
    transaction_sum_formatted = "{:.2f}".format(transaction_sum)

    # Aggregation for reported stray animals
    yearly_reported_pipeline = [
        {'$addFields': {'date': {'$toDate': '$date'}}},
        {'$group': {'_id': {'year': {'$year': '$date'}}, 'count': {'$sum': '1'}}},
        {'$sort': {'_id.year': 1}}
    ]
    monthly_reported_pipeline = [
        {'$addFields': {'date': {'$toDate': '$date'}}},
        {'$group': {'_id': {'year': {'$year': '$date'}}, 'month': {'$month': '$date'}}, 'count': {'$sum': '1'}}},
        {'$sort': {'_id.year': 1, '_id.month': 1}}
    ]
    yearly_reported_counts = list(report_collection.aggregate(yearly_reported_pipeline))
    monthly_reported_counts = list(report_collection.aggregate(monthly_reported_pipeline))

    # Aggregation for saved stray animals
    yearly_saved_pipeline = [
        {'$addFields': {'date': {'$toDate': '$date_of_intake'}}},
        {'$group': {'_id': {'year': {'$year': '$date'}}, 'count': {'$sum': '1'}}},
        {'$sort': {'_id.year': 1}}
    ]
    monthly_saved_pipeline = [
        {'$addFields': {'date': {'$toDate': '$date_of_intake'}}},
        {'$group': {'_id': {'year': {'$year': '$date'}}, 'month': {'$month': '$date'}}, 'count': {'$sum': '1'}}},
        {'$sort': {'_id.year': 1, '_id.month': 1}}
    ]
    yearly_saved_counts = list(animal_collection.aggregate(yearly_saved_pipeline))
    monthly_saved_counts = list(animal_collection.aggregate(monthly_saved_pipeline))

    # Aggregation for adopted animals
    yearly_adopted_pipeline = [
        {'$match': {'adoption_status': 'fee_paid'}},
        {'$addFields': {'date': {'$toDate': '$submission_datetime'}}},
        {'$group': {'_id': {'year': {'$year': '$date'}}, 'count': {'$sum': '1'}}},
        {'$sort': {'_id.year': 1}}
    ]
    monthly_adopted_pipeline = [
        {'$match': {'adoption_status': 'fee_paid'}},
        {'$addFields': {'date': {'$toDate': '$submission_datetime'}}},
        {'$group': {'_id': {'year': {'$year': '$date'}}, 'month': {'$month': '$date'}}, 'count': {'$sum': '1'}}},
        {'$sort': {'_id.year': 1, '_id.month': 1}}
    ]
    yearly_adopted_counts = list(adoption_collection.aggregate(yearly_adopted_pipeline))
    monthly_adopted_counts = list(adoption_collection.aggregate(monthly_adopted_pipeline))

    # Aggregation for transaction amounts
    yearly_transaction_pipeline = [
        {'$addFields': {'date': {'$toDate': '$date'}, 'amount': {'$toDouble': '$amount'}}},
        {'$group': {'_id': {'year': {'$year': '$date'}}, 'total_amount': {'$sum': '$amount'}}},
        {'$sort': {'_id.year': 1}}
    ]
    monthly_transaction_pipeline = [
        {'$addFields': {'date': {'$toDate': '$date'}, 'amount': {'$toDouble': '$amount'}}},
        {'$group': {'_id': {'year': {'$year': '$date'}}, 'month': {'$month': '$date'}}, 'total_amount': {'$sum': '$amount'}}},
        {'$sort': {'_id.year': 1, '_id.month': 1}}
    ]
    yearly_transaction_counts = list(transaction_collection.aggregate(yearly_transaction_pipeline))
    monthly_transaction_counts = list(transaction_collection.aggregate(monthly_transaction_pipeline))

    # Aggregation for reported animal types
    reported_animal_type_distribution_pipeline = [
        {'$group': {'_id': '$animal_type', 'count': {'$sum': '1'}}},
        {'$sort': {'count': -1}}
    ]
    reported_animal_type_distribution = list(report_collection.aggregate(reported_animal_type_distribution_pipeline))

    # Aggregation for adopted animal types
    adopted_animal_type_distribution_pipeline = [
        {'$match': {'adoption_status': 'fee_paid'}}, # Filter adopted animals
        {'$lookup': {
            'from': 'animal',
            'localField': 'animal_id',
            'foreignField': 'animal_id', # Ensure 'foreignField' matches 'localField'
            'as': 'reported_animal'
        }},
        {'$unwind': '$reported_animal'}, # Unwind to normalize the joined data
        {'$group': {'_id': '$reported_animal.species', 'count': {'$sum': '1'}}}, # Group by animal type
        {'$sort': {'count': -1}} # Sort by count in descending order
    ]
    adopted_animal_type_distribution = list(adoption_collection.aggregate(adopted_animal_type_distribution_pipeline))

    # Aggregation for saved animal types
    saved_animal_type_distribution_pipeline = [
        {'$group': {'_id': '$species', 'count': {'$sum': '1'}}},
        {'$sort': {'count': -1}}
    ]
    saved_animal_type_distribution = list(animal_collection.aggregate(saved_animal_type_distribution_pipeline))

    return render_template('analysis.html',
        reported_stray_count=reported_stray_count,
        saved_stray_count=saved_stray_count,
        adopted_count=adopted_count,
        transaction_sum=transaction_sum_formatted,
        yearly_reported_counts=yearly_reported_counts,
        monthly_reported_counts=monthly_reported_counts,
        yearly_saved_counts=yearly_saved_counts,
        monthly_saved_counts=monthly_saved_counts,
        yearly_adopted_counts=yearly_adopted_counts,
        monthly_adopted_counts=monthly_adopted_counts,
        yearly_transaction_counts=yearly_transaction_counts,
        monthly_transaction_counts=monthly_transaction_counts,
        reported_animal_type_distribution=reported_animal_type_distribution,
        adopted_animal_type_distribution=adopted_animal_type_distribution,
        saved_animal_type_distribution=saved_animal_type_distribution)

```

Figure 5.3.3.2 Analysis Complex Queries

5.3.4 Database Deployment

Step 1: Create a MongoDB Atlas account

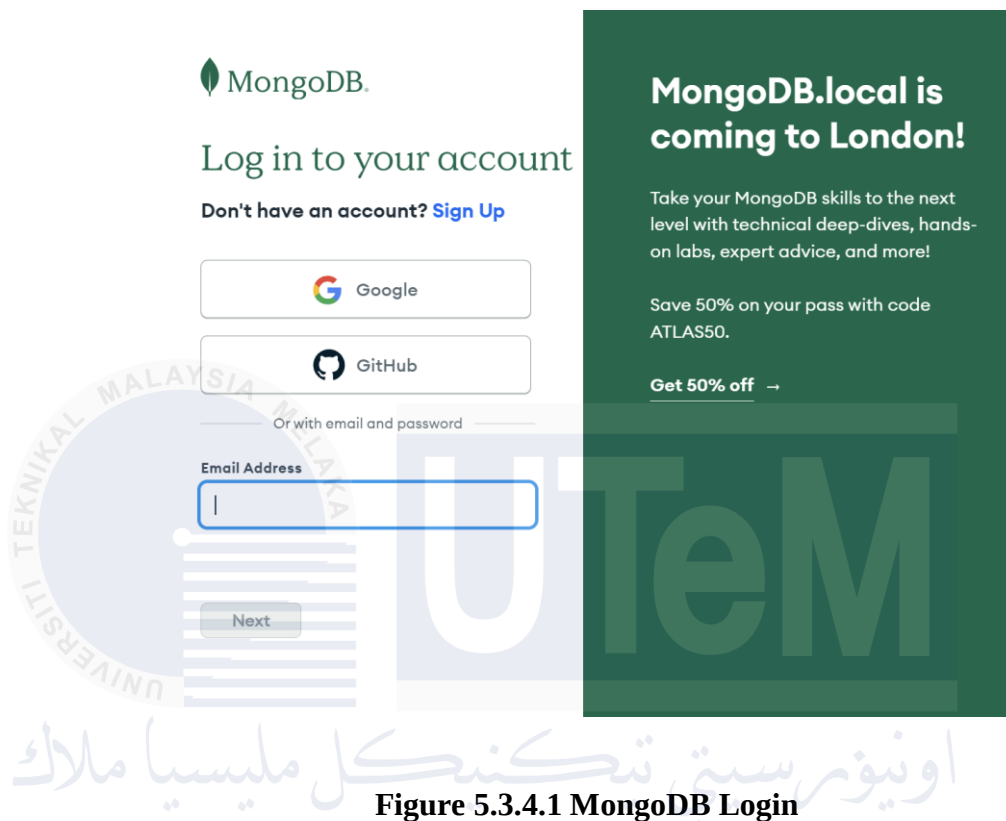


Figure 5.3.4.1 MongoDB Login

Step 2: Click the free plan and select provider. Then, choose the region and provide the name for the cluster.

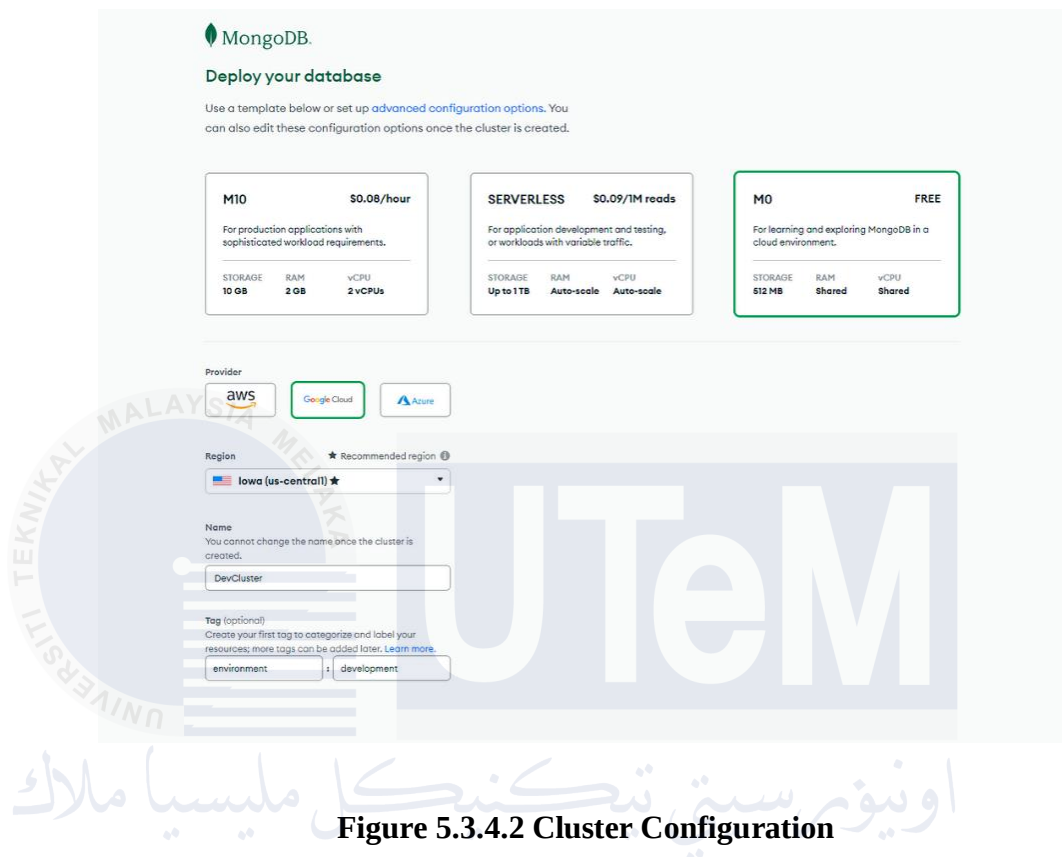


Figure 5.3.4.2 Cluster Configuration

Step 3: Go to the security section, create a user and password that can access the cluster.

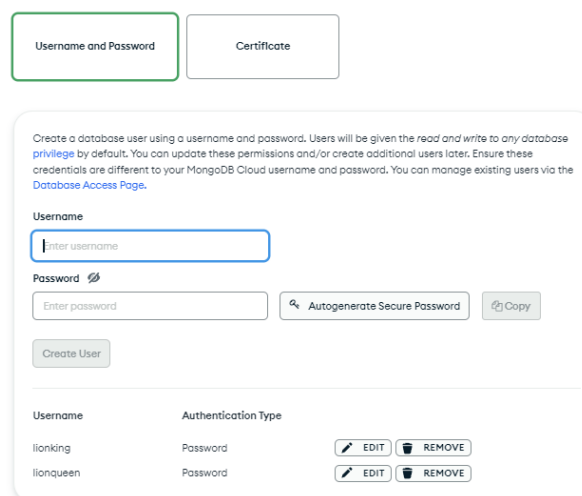


Figure 5.3.4.3 User Configuration

Step 4: Add the Ip Address of the device that will connect to the cluster.

✓ Where would you like to connect from?

Enable access for any network(s) that need to read and write data to your cluster.

My Local Environment

Use this to add network IP addresses to the IP Access List. This can be modified at any time.

ADVANCED

Cloud Environment

Use this to configure network access between Atlas and your cloud or on-premise environment. Specifically, set up IP Access Lists, Network Peering, and Private Endpoints.

Add entries to your IP Access List

Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the [Network Access Page](#).

IP Address	Description	
Enter IP Address	Enter description	Add My Current IP Address
Add Entry		

Figure 5.3.4.4 IP access list

Step 5: Click connect at the dashboard.

Clusters

Find a database deployment...

Cluster0

Visualize Your Data

Build dashboards and charts, and embed them in your apps with MongoDB Charts.

● R 0.07
 ● W 0

Last 12 minutes
0.07%

?

Figure 5.3.4.5 Cluster Dashboard

Step 6: Choose Compass.

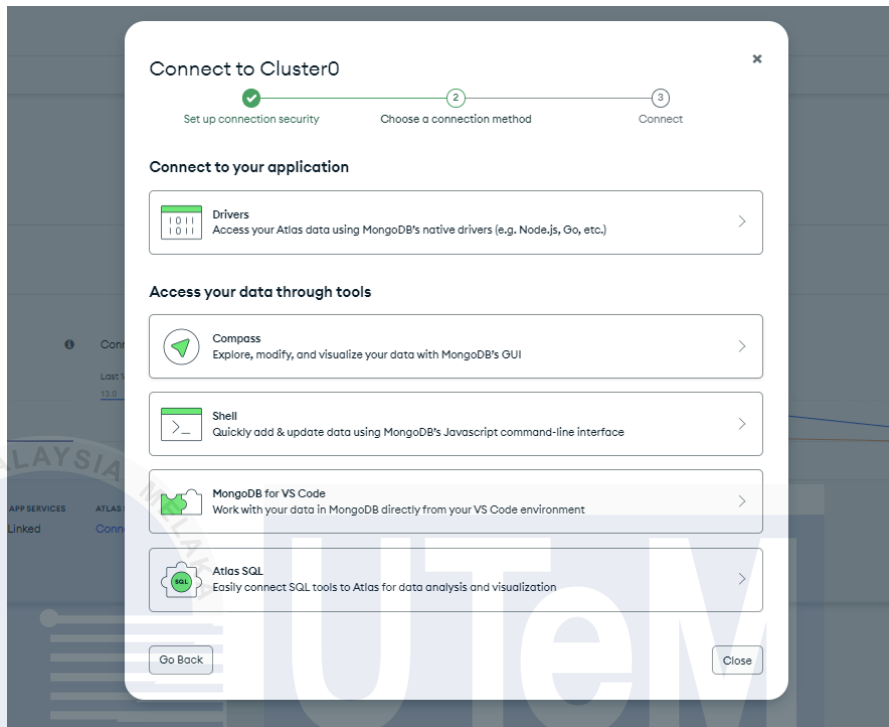


Figure 5.3.4.6 Cluster Connection

Step 7: Copy the connection string.

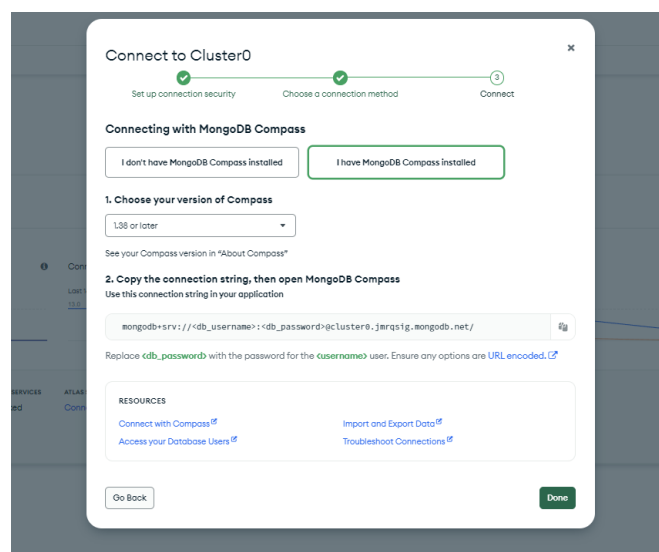


Figure 5.3.4.7 Connection String setup

Step 8: Paste the connection String in MongoDB Compass and input the username and password.

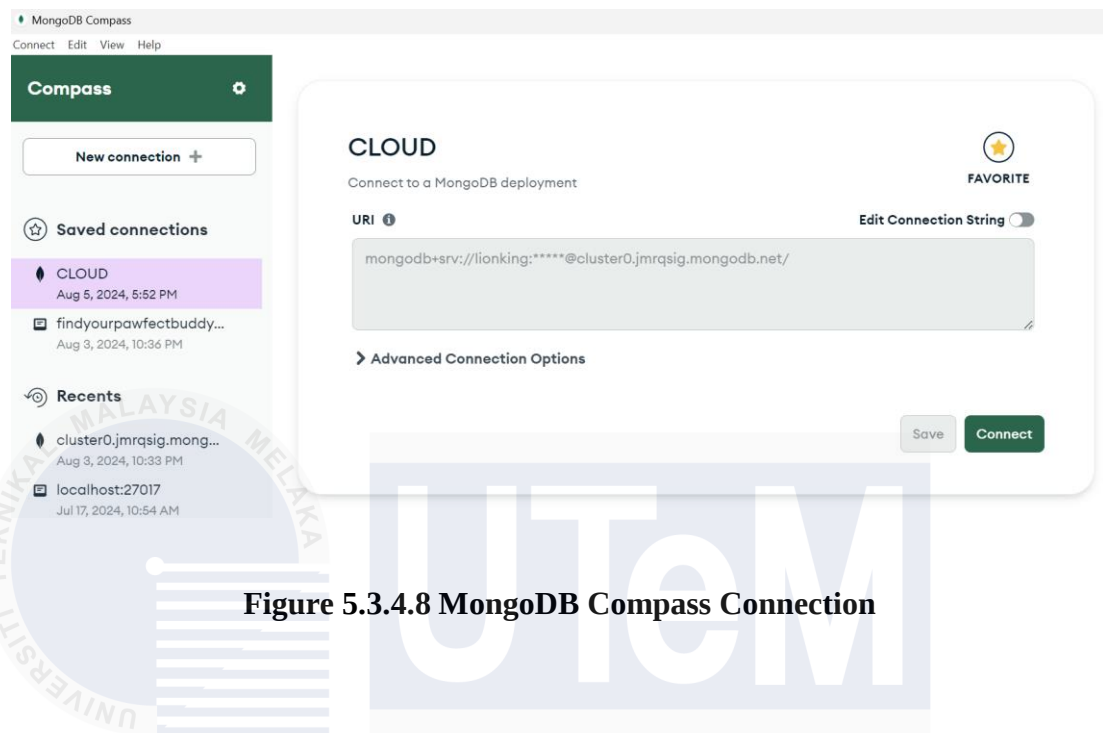


Figure 5.3.4.8 MongoDB Compass Connection

Step 9: Provide database name and click create database.

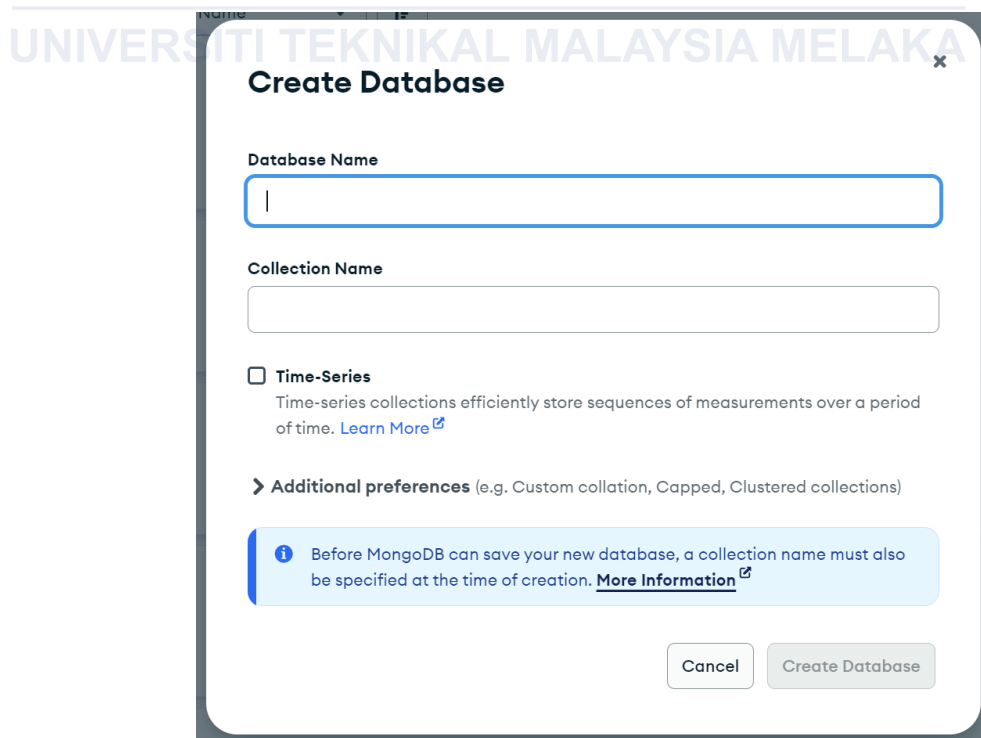


Figure 5.3.4.9 Create Database

Step 10: Click add collection and provide the collection name.

Create Collection

Collection Name

☐ **Time-Series**
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> **Additional preferences** (e.g. Custom collation, Capped, Clustered collections)

Cancel Create Collection

Documents:	Avg. document size:	Indexes:
11	26.84 kB	1

Figure 5.3.4.10 Create Collection

Step 11: Add the data by inserting document.

Insert Document

To collection user_database.adoption

VIEW

```

1  /**
2  * Paste one or more documents here
3  */
4  {
5    "_id": {
6      "$oid": "66c6e2834ab895931efbbe91"
7    }
8  }

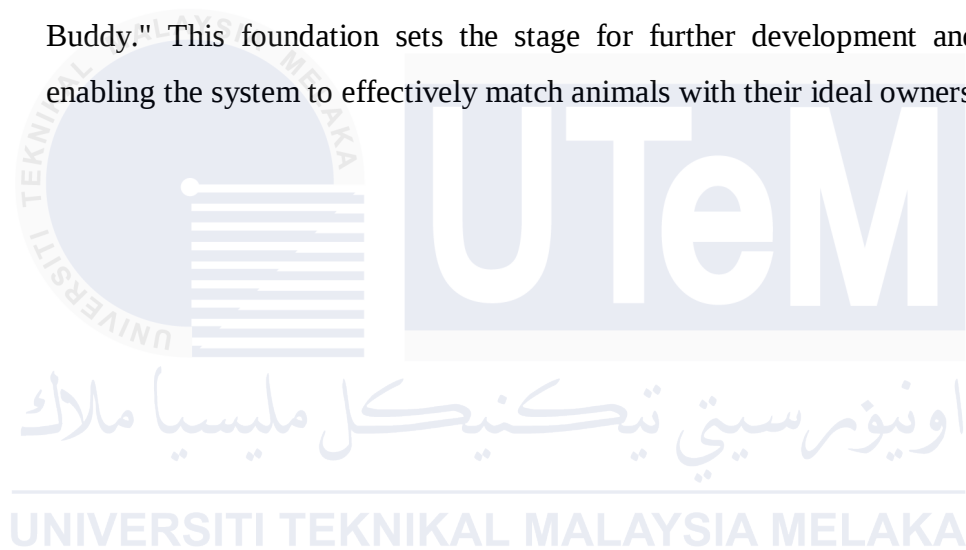
```

Cancel Insert

Figure 5.3.4.11 Insert Document

5.4 Conclusion

In summary, this chapter provides comprehensive guidance on setting up the MongoDB environment for the "Find Your Pawfect Buddy" system. It covers the necessary steps to install MongoDB on various platforms and provides detailed instructions for creating collections and inserting data specific to the system's needs. The database implementation focuses on managing the adoption process, animal details, medical records, memorable moments, reports, staff information, transactions, and user data. By utilizing MongoDB's capabilities, we ensure a robust and efficient management system that supports the seamless operation of "Find Your Pawfect Buddy." This foundation sets the stage for further development and integration, enabling the system to effectively match animals with their ideal owners.



CHAPTER 6: TESTING

6.1 Introduction

Testing is the process of assessing a software application's functionality to determine whether the developed software complied with the requirements and to find errors to guarantee that the system is error-free. In this chapter, verification and validation will be done on "Find Your Pawfect Buddy", a stray animal reporting and adoption system.

The "Find Your Pawfect Buddy" testing has two primary objectives:

- i. To show end users that "Find Your Pawfect Buddy" satisfies their user requirements
- ii. To find any errors or bugs in the "Find Your Pawfect Buddy" using a different test approach

System testing is also a crucial stage in the Database Life Cycle (DBLC). The test plan, which includes the test environment, schedule, and organization, is what makes up the "Find Your Pawfect Buddy" testing phase.

6.2 Test Plan

A test plan is a technical document that outlines the test strategy, scope, schedule, resources needed for the testing process, and test deliverables. A thorough understanding of the procedures and features of the test plan is included in the system and outlines the testing procedures for each of those to detect bugs, ascertain the system's true limitations, and ascertain whether the system functions as intended.

Test Organization

In the "Find Your Pawfect Buddy", there are three primary users: Adopters, Public Users, and Animal Shelters Staff. Functional requirements and non-requirements will be tested on each user role. In Table 6.2.1.1, the table illustrates how each user undergoes testing based on their responsibilities.

Table 5.3.4.1 User Responsibilities List

Tester ID	Users	Responsibilities
T1	Adopter	• Testing the system with the adoption process • Verifying the adoption application functionality • Defect and bug detection
T2	Public Users	• Testing the system with the reporting process • Verifying the reporting module functionality • Defect and bug detection
T3	Animal Shelters Staff	• Testing the system with the animal showcase and management processes • Verifying the animal management module functionality • Defect and bug detection

6.2.1 Test Environment

The gear, software, operating system, necessary tools, and network configuration that the testing teams need to run test cases are all set up in the test environment. The list of hardware and software needed for the "Find Your Pawfect Buddy" test environment is displayed in tables 6.2.2.1 and table 6.2.2.2.

Table 6.2.1.1 Test Environment Hardware List

Environment Specification	Description
Laptop	Asus ROG Zephyrus G15
Processor	AMD Ryzen 9 5900HS
Keyboard and Mouse	Logitech
Random Access Memory (RAM)	16GB

Table 6.2.1.2 Test Environment Software List

Environment	Description
Database	MongoDB To manage data in the database table that runs on a server
Operating System/Platform	Window 11 To manage the resources of computers, both hardware and software, and then offering the service or tool needed for computer programs to run
Web Browser	Google Chrome To use run the python flask source code and test the system interface functionality
Microsoft Word 2013 / Power Point 2013	To use write final report and make a power point slide for presentation

6.2.2 Test Schedule

A test schedule outlines the plan and key checkpoints for testing. This schedule depends on when the system is fully tested and ready. To create the test schedule, estimated completion dates should be determined and adjusted as needed. Table 6.2.3.1 shows the planned test schedule for the "Find Your Pawfect Buddy" project.

Table 6.2.2.1 Test Schedule

Testing Task	Testing Activity	Start Date	End Date
Stray Animal Reporting	Unit Testing, Integration Testing	29-7-2024	31-7-2024
Staff Account Management	Unit Testing, Integration Testing	1-8-2024	2-8-2024
Adoption Management	Unit Testing, Integration Testing	3-8-2024	5-8-2024
Analysis Reporting	Unit Testing, Integration Testing	6-8-2024	7-8-2024
Adopter Account Management	Unit Testing, Integration Testing	8-8-2024	9-8-2024
Adopting Animal	Unit Testing, Integration Testing	10-8-2024	11-8-2024

6.3 Test Strategy

A test strategy is a set of guidelines that outlines the design of tests, the techniques to be used, and which modules need to be tested. It ensures that the testing process is systematic and comprehensive.

White Box Testing on the other hand, requires knowledge of the internal structure and code of the application. This type of testing involves checking the internal logic, code structure, and branches of the software. It is generally performed by developers who have access to the source code. White Box Testing is most commonly applied in unit testing and integration testing to ensure that the internal workings of the application are functioning correctly.

Unit Testing is a level of software testing where individual units or components of the software are tested separately. The goal is to validate that each unit performs its intended function as designed. For "Find Your Pawfect Buddy," unit testing would involve testing specific modules such as user login or pet matching functionality to ensure each module works correctly on its own.

System Testing is conducted on the complete and integrated software to verify that it meets the specified requirements. For "Find Your Pawfect Buddy," system testing would ensure that the entire application flows smoothly from user registration to pet matching and that all components work together as intended.

Table 6.2.2.1 Type of test and test design techniques for white box and black box testing

Aspect	White Box Testing
Type of Test	Tests internal structures and code.
Test Design Techniques	• Code Coverage: Ensures all code paths are executed. • Path Testing: Checks all possible paths through the code. • Branch Testing: Validates each branch of the code. • Statement Testing: Ensures that each statement in the code is executed.
Purpose	To verify the correctness of code logic, flow, and structure.
Typical Tests	Unit testing, Integration testing.
Knowledge Required	Requires detailed knowledge of the code and its structure.
Focus	Code correctness, internal logic, and program flow.
Testing Level	Mostly used in low-level testing such as unit and integration testing.

6.3.1 Classes of Tests

The following describes several types of test classes implemented for "Find Your Pawfect Buddy":

I. Error Handling Test

This test validates that only correct and accurate data can be entered into input fields. For example, users should only be able to enter numbers into the "Phone No." field. In "Find Your Pawfect Buddy," this test ensures that users provide valid data before storing it in the database. If invalid or null input is detected, an error message is displayed to inform the user.

II. Security Test

Security testing is conducted to verify the authentication process, including the validation of user email and password during login. In "Find Your Pawfect Buddy," this test ensures that login credentials are properly verified to protect user accounts and personal information.

III. Integration Test

Integration testing ensures that data is correctly captured and stored in the database according to user inputs. This involves navigating through each menu item in the interface to ensure that data transfer between modules is accurate. For example, only users who complete all necessary steps will be added to the recruitment list.

6.4 Test Design

Test design involves creating and writing test suites to ensure that software requirements are met according to client needs. It includes two main components: test description and test data.

6.4.1 Test Description

Test descriptions outline the identification of test cases, types of testing, pre-conditions, test requirements, procedures, and expected results. Each module test case is documented with detailed descriptions. Tables 6.4.1.1 through 6.4.1.23 provide detailed test descriptions according to the system modules.

Table 6.4.1.1 Test Description of User Login

Test ID	T001- User Login			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Security and error handling testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC1_1	Validate the login function is available if the username and password provide are valid	User has valid username and password	1. Navigate to login page 2. Provide valid username 3. Provide valid password 4. Click on Login button	Login successful
TC1_2	Validate the login function is Not available if the username or password are blank		1. Navigate to login page 2. Click on Login button	Login failed. Display error message “Please fill out this field”
TC1_3	Validate the login function is Not available if the username and password provide are invalid		1. Navigate to login page 2. Provide valid username 3. Provide invalid password	Login failed. Display error message “Invalid username or Password!”

			4. Click on Login button	
--	--	--	--------------------------	--

Table 6.4.1.2 Test Description of Browse Animals

Test ID	T002- Browse Animals			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functional and UI testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC2_1	Validate that the filter form allows filtering by species.	The page loads with available animal data.	1. Navigate to the Browse Animals page 2. Select "Dog" from the Species dropdown 3. Click on "Apply Filters"	The page refreshes showing only animals with species "Dog".
TC2_2	Validate that the filter form allows filtering by age.	The page loads with available animal data.	1. Navigate to the Browse Animals page 2. Select "Young" from the Age dropdown 3. Click on "Apply Filters"	The page refreshes showing only young animals.

TC2_3	Validate that the search function filters animals by name or description.	The page loads with available animal data.	1. Navigate to the Browse Animals page 2. Enter "Dog1" in the search field 3. Click on "Apply Filters"	The page refreshes showing animals with names or descriptions containing " Dog1".
TC2_4	Validate that the reset button clears all filters and reloads the page with all animals.	The page loads with applied filters.	1. Navigate to the Browse Animals page 2. Apply any filters 3. Click on "Reset Filters"	The page reloads showing all animals, with no filters applied.

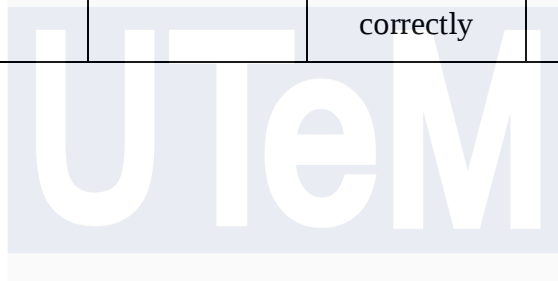
اونیورسیتی تکنیکل ملیسیا ملاک

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Table 6.4.1.3 Test Description of Animal Details Page

Test ID	T003 - Animal Details Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	UI functionality and user interaction testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC3_1	Validate the Animal Details page loads correctly when a valid animal ID is provided	Animal ID is valid and exists in the database	1. Navigate to the Animal Details page with a valid animal ID 2. Observe if the animal details are correctly displayed	Animal details (photo, name, species, breed, etc.) are displayed correctly
TC3_2	Validate the "Schedule Adoption Meeting" modal appears when the button is clicked	Animal ID is valid and exists in the database	1. Navigate to the Animal Details page 2. Click on the "Schedule Adoption Meeting" button	Modal appears with date/time fields and a submit button
TC3_3	Validate that the modal can be closed by clicking the 'x' button or	"Schedule Adoption Meeting" modal is open	1. Open the "Schedule Adoption Meeting" modal 2. Click the 'x'	Modal closes successfully

	outside the modal area		button 3. Click outside the modal	
TC3_4	Validate that the animal moment records are displayed if they exist	Animal has moment records in the database	1. Navigate to the Animal Details page 2. Observe if the moment records are displayed correctly	Moment records with photos and captions are displayed



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Table 6.4.1.4 Test Description of Report Stray Animal Form

Test ID	T004 - Report Stray Animal Form			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Form validation and data submission testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC4_1	Validate the form submission is successful when all fields are correctly filled	All form fields have valid inputs	1. Navigate to the report form page 2. Select "Dog" for animal type 3. Select a location on the map 4. Enter a description 5. Enter a valid phone number 6. Upload a photo 7. Click on Submit button	Form submitted successfully and data is saved

TC4_2	Validate form submission fails if no location is selected	No location selected on the map	1. Navigate to the report form page 2. Select "Cat" for animal type 3. Do not select a location on the map 4. Enter a description 5. Enter a valid phone number 6. Upload a photo 7. Click on Submit button	Submission fails. Display error message "Please select a location on the map."
TC4_3	Validate form submission fails if animal type is not selected	Animal type is not selected	1. Navigate to the report form page 2. Do not select an animal type 3. Select a location on the map 4. Enter a description 5. Enter a valid phone number 6. Upload a photo 7. Click	Submission fails. Display error message "Please select either a dog or a cat."

			on Submit button	
--	--	--	---------------------	--

Table 6.4.1.5 Test Description of Appointment Page

Test ID	T005 – Appointment Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	UI and functional testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC5_1	Validate that the appointment information is displayed correctly when appointments exist.	There are existing appointments in the system.	1. Navigate to the Appointment Information page. 2. Ensure that there are appointments in the database.	The page displays appointment details in a table format.
TC5_2	Validate that the message "No appointments found." is displayed when no appointments exist.	No appointments exist in the system.	1. Navigate to the Appointment Information page. 2. Ensure that there are no appointments in the database.	The page displays the message "No appointments found."

TC5_3	Validate that the "Accept Meet Up" and "Reject Meet Up" buttons are displayed for appointments with the status "meet_up".	There is an appointment with the status "meet_up".	1. Navigate to the Appointment Information page. 2. Ensure that there is an appointment with the status "meet_up".	"Accept Meet Up" and "Reject Meet Up" buttons are displayed.
TC5_4	Validate that the "Meeting Location" field is displayed only if the adoption status is not "apply".	Appointment exists with a status other than "apply".	1. Navigate to the Appointment Information page. 2. Ensure that the adoption status is not "apply".	The "Meeting Location" field is displayed.

Table 6.4.1.6 Test Description of Update User Details Page

Test ID	T006 – Update User Details Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	UI and functional testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC6_1	Validate that the update function is available if all fields are filled correctly	User is logged in	1. Navigate to the update details page 2. Fill all fields correctly 3. Provide valid old password 4. Click on Update button	Update successful. Show "Update successful!" message
TC6_2	Validate that the update function is Not available if the old password is blank		1. Navigate to the update details page 2. Leave the old password field blank 3. Click on Update button	Update failed. Display error message "Old password is required."
TC6_3	Validate that the update function is Not available if new password		1. Navigate to the update details page 2. Provide valid old	Update failed. Display error message "New password and confirm password do not match."

	and confirm password do not match		password 3. Provide non-matching new password and confirm password 4. Click on Update button	
--	-----------------------------------	--	--	--

Table 6.4.1.7 Test Description of Staff Login Page

Test ID	T007 - Staff Login Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Security and error handling testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC7_1	Validate the login function is available if the staff ID and password provided are valid	User has valid staff ID and password	1. Navigate to the login page 2. Provide valid staff ID 3. Provide valid password 4. Click on the Login button	Login successful
TC7_2	Validate the login function is not available if the staff ID or password is blank		1. Navigate to the login page 2. Click on the Login button	Login failed. Display error message “Please fill out this field”

TC7_3	Validate the login function is not available if the staff ID and password provided are invalid		1. Navigate to the login page 2. Provide valid staff ID 3. Provide invalid password 4. Click on the Login button	Login failed. Display error message "Invalid staff ID or Password!"
-------	--	--	---	--

Table 6.4.1.8 Test Description of Latest Report Page

Test ID	T008 – Latest Report Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functionality and UI testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC8_1	Validate that filtering reports by animal type works correctly.	Reports available with different animal types	1. Navigate to the report page. 2. Select "Dog" in the "Filter Animal" dropdown. 3. Click "Filter".	The report table displays only reports with animal type "Dog".
TC8_2	Validate that filtering reports	Reports available with	1. Navigate to the report page.	The report table displays only reports

	by status works correctly.	different statuses	2. Select "Checked" in the "Filter Status" dropdown. 3. Click "Filter".	with the status "Checked".
TC8_3	Validate that filtering reports by date range works correctly.	Reports available with different dates	1. Navigate to the report page. 2. Select a start date. 3. Select an end date. 4. Click "Filter".	The report table displays only reports within the selected date range.
TC8_4	Validate that the status of a report can be updated.	Reports available with editable status	1. Navigate to the report page. 2. Change the status of a report using the dropdown. 3. Observe the system response.	The status of the report is updated and a confirmation message "Status updated successfully" is displayed.
TC8_5	Validate that resetting the filters clears all applied filters and search inputs.	Filters and search are applied	1. Navigate to the report page. 2. Apply any filter or search. 3. Click the	All filters and search inputs are cleared, and the report table displays all reports without any filtering.

			"Clear" button.	
--	--	--	--------------------	--

Table 6.4.1.9 Test Description of Register New Staff Page

Test ID	T009 - Register New Staff Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Security and error handling testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC9_1	Validate the registration function is available if the staff ID and password provided are valid	User has valid staff ID and password	1. Navigate to the registration page 2. Provide valid staff ID 3. Provide valid password 4. Click on the Register button	Registration successful
TC9_2	Validate the registration function is not available if the staff ID or password are blank		1. Navigate to the registration page 2. Click on the Register button	Registration failed. Display error message "Please fill out this field"
TC9_3	Validate the registration		1. Navigate to the	Registration failed. Display error message

	function is not available if the staff ID or password provided are invalid		registration page 2. Provide invalid staff ID 3. Provide invalid password 4. Click on the Register button	"Invalid staff ID or Password!"
--	--	--	--	---------------------------------

Table 6.4.1.10 Test Description of Update Staff Details Page

Test ID	T010 - Update Staff Details Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Security and error handling testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC10_1	Validate that updating staff details is successful with valid inputs	User has valid existing details	1. Navigate to the update details page.	Validate that updating staff details is successful with valid inputs
TC10_2	Validate that the update fails when the old password is not provided		1. Navigate to the update details page. 2. Leave the old password	Update failed. Display error message "Old password is required."

			field blank. 3. Click on the "Update" button.	
TC10_3	Validate that the update fails when new passwords do not match		1. Navigate to the update details page. 2. Provide correct old password. 3. Provide non-matching new passwords. 4. Click "Update."	Update failed. Display error message "Passwords do not match."
TC10_4	Validate that deleting a staff account shows a confirmation dialog		1. Navigate to the update details page. 2. Click on "Delete Staff" button.	Display confirmation dialog.
TC10_5	Validate that staff account is deleted after confirmation		1. Follow steps in TC10_4. 2. Confirm the deletion in the dialog.	Staff account deleted successfully. Display success message.

Table 6.4.1.11 Test Description of View Animal Page

Test ID	T011 – View Animal Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Error handling testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC11_1	Validate that the search function allows valid data submission	User is on the view animal page.	1. Navigate to the view animal page. 2. Provide valid data in the search bar.	The animal list is displayed according to the search criteria.
TC11_2	Validate that the search function handles missing required fields	User is on the view animal page.	1. Navigate to the view animal page. 2. Leave the search bar empty.	Submission is blocked; error messages are displayed for missing fields.
TC11_3	Validate filtering by overall health status	Health status filter is visible	1. Navigate to the animal view page. 2. Select a health status "Good" from the health status filter dropdown. 3. Observe the filtered results.	Only animals with the selected health status "Good" are displayed in the table (e.g., Max and Charlie).

TC11_4	Validate filtering by vaccination status	Vaccination filter is visible	1. Navigate to the animal view page. 2. Select a vaccination status from the vaccination filter dropdown. 3. Observe the filtered results.	Only animals with the selected vaccination status are displayed in the table.
TC11_5	Validate filtering by spayed/neutered status	Spayed/Neutered filter is visible	1. Navigate to the animal view page. 2. Select a spayed/neutered status from the spayed/neutered filter dropdown. 3. Observe the filtered results.	Only animals with the selected spayed/neutered status are displayed in the table.
TC11_6	Validate date range filtering functionality	Date inputs are visible	1. Navigate to the animal view page. 2. Select a date range in the start date and end date inputs. 3. Observe the filtered results.	Only animals within the selected date range are displayed in the table.
TC11_7	Validate reset filters button functionality	Filters are applied	1. Navigate to the animal view page. 2. Apply	All filters are cleared and the full list of animals is displayed.

			various filters. 3. Click the "Clear" button. 4. Observe that all filters are reset and all animals are displayed.	
TC11_8	Validate "Add Animal" button functionality	Add Animal button is visible	1. Navigate to the animal view page. 2. Click the "Add Animal" button. 3. Observe redirection to the animal profile page.	User is redirected to the animal profile page.

Table 6.4.1.12 Test Description of Animal Profile Update Page

Test ID	T012 - Animal Profile Update Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functionality and UI Testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC12_1	Validate that the profile update function allows valid data submission	User is on the animal profile update page with valid data	1. Navigate to the animal profile update page. 2. Provide valid data in all required fields. 3. Click on the "Update Profile" button.	Profile is updated successfully; user is redirected or shown a confirmation message.
TC12_2	Validate that the profile update function prevents submission with missing required fields	User is on the animal profile update page with missing required data	1. Navigate to the animal profile update page. 2. Leave one or more required fields empty. 3. Click on the "Update Profile" button.	Submission is blocked; error messages are displayed for missing fields.

Table 6.4.1.13 Test Description of Add Animal Page

Test ID	T013 - Add Animal Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Form validation and error handling			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC13_1	Validate that the 'Add Animal' form page loads correctly	User is logged in	1. Navigate to the 'Add Animal' page. 2. Check if all fields (photo, name, species, breed, age, overall health, etc.) are present.	All form fields are displayed correctly and ready for input.
TC13_2	Validate that the form can be submitted when all required fields are filled	All required fields filled	1. Fill out the form with valid data for all required fields. 2. Click the 'Submit' button.	Form is submitted successfully, and the user is redirected to the next page
TC13_3	Validate that the form cannot be submitted when a	Any required field is blank	1. Leave one or more required fields blank. 2. Click the	Submission fails, and an error message appears indicating missing fields.

	required field is left blank		'Submit' button.	
TC13_4	Validate that the hidden date field is automatically populated with the current date	Page is loaded	1. Load the 'Add Animal' page. 2. Inspect the value of the hidden date field.	Date field is automatically set to the current date.
TC13_5	Validate that the form does not accept invalid file types for the photo upload	User attempts to upload a photo	1. Attempt to upload a file that is not an image (e.g., .txt or .pdf) in the photo field. 2. Click the 'Submit' button.	Form submission is prevented, and an error message appears.

Table 6.4.1.14 Test Description of Medical Record Page

Test ID	T014 - Medical Record Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functional testing and UI testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC14_1	Validate the "Medical Record" page displays correctly for a valid animal_id	The animal_id is valid and has associated treatment records	1. Navigate to the "Medical Record" page 2. Verify the page displays treatment records	The page displays the correct treatment records for the animal_id.
TC14_2	Validate the "Medical Record" page shows a message when no records are available	The animal_id is valid but has no associated treatment records	1. Navigate to the "Medical Record" page 2. Verify the "No treatment records found for this animal." message is displayed	The page displays "No treatment records found for this animal."
TC14_3	Validate the "Add Treatment" button redirects to the correct page	The animal_id is valid	1. Navigate to the "Medical Record" page 2. Click on the "Add Treatment" button	The page redirects to the new treatment page for the specific animal_id.

			3. Verify the redirection to the new treatment page	
TC14_4	Validate the "Edit" button redirects to the correct page for treatment	The animal_id and treatment_id are valid	1. Navigate to the "Medical Record" page 2. Click on the "Edit" button for a treatment record 3. Verify the redirection to the treatment editing page	The page redirects to the treatment editing page for the specific treatment_id.

Table 6.4.1.15 Test Description of Add Treatment Page

Test ID	T015 - Add Treatment Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functional and error handling testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC15_1	Validate that the form is displayed correctly and fields are available for input	The 'Add Treatment' page is loaded	1. Navigate to the 'Add Treatment' page 2. Verify the presence of all input fields	Form displays correctly with all fields

			and dropdowns	
TC15_2	Validate the form submission with valid inputs	All required fields are filled with valid data	1. Fill in all the required fields 2. Click on the 'Submit' button	Treatment details are submitted successfully
TC15_3	Validate that an error is displayed if required fields are left blank		1. Leave one or more required fields blank 2. Click on the 'Submit' button	Error message displayed indicating required fields must be filled
TC15_4	Validate the dynamic population of treatment names based on selected treatment type	The 'Add Treatment' page is loaded	1. Select a treatment type from the dropdown 2. Verify the population of the treatment names dropdown	Treatment names are populated correctly based on the treatment type
TC15_5	Validate the correct handling of the 'Others' option	The 'Add Treatment' page is loaded	1. Select 'Others' in the treatment type dropdown 2. Verify that	Text input for treatment name appears when 'Others' is selected

	in the treatment type		the text input for the treatment name appears	
--	--------------------------	--	--	--

Table 6.4.1.16 Test Description of Update Treatment Page

Test ID	T016- Update Treatment Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functionality, Validation, and Error Handling Testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC16_1	Validate the update function works with all required fields provided	Treatment ID exists in the database	1. Navigate to the Update Treatment page 2. Provide valid Animal ID 3. Provide valid Treatment Date 4. Select valid Treatment Type and Treatment Name 5. Click on Update button	Update successful. The treatment record is updated in the database.

TC16_2	Validate that an error is shown if required fields are missing	Treatment ID exists in the database	1. Navigate to the Update Treatment page 2. Leave required fields like Treatment Date empty 3. Click on Update button	Update failed. Display error message "Please fill out this field"
TC16_3	Validate that the delete function works as expected	Treatment ID exists in the database	1. Navigate to the Update Treatment page 2. Click on the Delete button	Delete successful. The treatment record is removed from the database.

Table 6.4.1.17 Test Description of View Moment Page

Test ID	T017 - View Moment Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functionality and UI testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC17_1	Validate that the moment records display correctly for an animal	Moments exist for the animal	1. Navigate to the Moment Record page. 2. Ensure that moments are displayed. 3. Verify the	Moments are displayed with correct photo, date, and caption.

			photo, date, and caption.	
TC17_2	Validate that the "Add Moment" button is functional		1. Navigate to the Moment Record page. 2. Click on the "Add Moment" button.	Redirect to the "Add Moment" page.
TC17_3	Validate that the "Edit Post" button is functional for each moment	Moments exist for the animal	1. Navigate to the Moment Record page. 2. Click on the "Edit Post" button for a specific moment.	Redirect to the "Edit Post" page with the correct moment data pre-filled.
TC17_4	Validate that a message is displayed when no moments are recorded	No moments exist for the animal	1. Navigate to the Moment Record page.	Display message "No moments recorded for this animal."

Table 6.4.1.18 Test Description of Add Moment Page

Test ID	T018 - Add Moment Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functional and validation testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC18_1	Validate that the animal ID displays correctly	Animal ID exists for the animal	1. Navigate to the add moment page. 2. Verify the animal ID is correct.	Animal ID is displayed automatically and correctly based on the chosen animal.
TC18_2	Validate form submission with missing photo	User is on the 'Add Moment' page	1. Enter a valid Animal ID. 2. Leave moment photo field empty. 3. Select a moment date. 4. Enter a caption. 5. Click on 'Upload Moment' button.	Form submission fails. Display error message: "Moment photo is required."
TC18_3	Validate form submission	User is on the 'Add Moment' page	1. Enter a valid Animal ID.	Form submission fails. Display error message:

	with missing moment date		2. Upload a valid photo. 3. Leave moment date field empty. 4. Enter a caption. 5. Click on 'Upload Moment' button.	"Moment date is required."
TC18_4	Validate form submission with missing caption	User is on the 'Add Moment' page	1. Enter a valid Animal ID. 2. Upload a valid photo. 3. Select a moment date. 4. Leave caption field empty. 5. Click on 'Upload Moment' button.	Form submission fails. Display error message: "Caption is required."
TC18_5	Validate successful form submission	Valid animal ID, photo, date, and caption provided	1. Enter a valid Animal ID. 2. Upload a valid photo. 3. Select a moment date.	Form is submitted successfully. User is redirected to a confirmation page or receives a success message.

			4. Enter a caption. 5. Click on 'Upload Moment' button.	
--	--	--	--	--

Table 6.4.1.19 Test Description of Update Moment Page

Test ID	T019 - Update Moment Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functionality and UI testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC19_1	Validate the form submission with valid data.	User is on the "Update Moment" page with a moment pre-loaded.	1. Enter valid data in all fields. 2. Click "Update Moment". 3. Verify successful update.	The moment is updated with the entered details, and a success message is displayed.
TC19_2	Validate form submission with missing required fields.	User is on the "Update Moment" page with a moment pre-loaded.	1. Leave one or more required fields empty. 2. Click "Update Moment". 3. Verify error	Appropriate error messages are displayed for missing fields, and the moment is not updated.

			messages are displayed.	
TC19_3	Validate photo upload functionality.	User is on the "Update Moment" page with a moment pre-loaded.	1. Select a new photo. 2. Click "Update Moment". 3. Verify the new photo is uploaded and displayed.	The new photo is uploaded and displayed correctly.
TC19_4	Validate the delete moment functionality.	User is on the "Update Moment" page with a moment pre-loaded.	1. Click "Delete Moment". 2. Confirm the deletion. 3. Verify the moment is removed from the list.	The moment is removed from the list, and a success message is displayed.

Table 6.4.1.20 Test Description of Adoption Page

Test ID	T020 – Adoption Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functional and validation testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC20_1	Validate that the adoption appointment status updates correctly when a new status is selected	Adoption exists	1. Navigate to the adoption management page 2. Select a new status from the dropdown menu 3. Click on the "Save" button	The status is updated in the database and redirected to payment page.
TC20_2	Validate that adoption appointment details (location and notes) are editable and saved correctly	Adoption exists	1. Navigate to the adoption management page 2. Edit the meeting location and note fields 3. Click on the "Save" button	Updated location and notes are saved and reflected in the UI
TC20_3	Validate that an adoption is deleted	Adoption exists	1. Navigate to the adoption management	The adoption entry is removed from the table and the database

	correctly upon clicking the delete button		page 2. Click on the "Delete" button next to an adoption 3. Confirm deletion in the prompt	
--	---	--	--	--

Table 6.4.1.21 Test Description of View Transaction Page

Test ID	T021 – View Transaction Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functional and validation testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC21_1	Validate the filter function by date range	Transactions are available in the database	1. Navigate to View Transactions page 2. Select a start date 3. Select an end date 4. Click on the Filter button	Transactions are filtered by the selected date range
TC21_2	Validate the filter function by payment method	Transactions are available in the database	1. Navigate to View Transactions page 2. Select a payment method from	Transactions are filtered by the selected payment method

			the dropdown 3. Click on the Filter button	
TC21_3	Validate the export to PDF function	Transactions are available in the database	1. Navigate to View Transactions page 2. Click on the "Export as PDF" button	PDF is generated and downloaded with the correct transaction data

Table 6.4.1.22 Test Description of Add Transaction Page

Test ID	T022 – Add Transaction Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functional and validation testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC22_1	Validate that all fields are displayed correctly in the form	User is logged in and has access to the add transaction page	1. Navigate to the "Add Transaction" page 2. Verify that all fields are displayed correctly	All fields (Adoption ID, Animal ID, Username, Date, Description, Amount, Payment Method) are displayed correctly with appropriate labels
TC22_2	Validate that the Date field is auto-populated with the current date	User is logged in and has access to the add transaction page	1. Navigate to the "Add Transaction" page 2. Verify that the Date field is auto-	The Date field is auto-populated with the current date

			populated with today's date	
TC22_3	Validate that the form submission works correctly when all mandatory fields are filled	User has valid inputs for all fields	1. Fill in all fields 2. Click the "Submit" button	The transaction is successfully added and the user is redirected to the confirmation page

Table 6.4.1.23 Test Description of Analysis Page

Test ID	T023 – Analysis Page			
Testing Type	Unit testing and integration testing			
Test Strategy	White Box Testing			
Test Class	Functional and validation testing			
Test Case ID	Test Requirements	Pre-condition	Test/Step Procedure	Expected Output
TC23_1	Validate that the map displays correctly with initial settings		1. Open the page. 2. Verify that the map loads with the default view.	Map is displayed with default view centered on Malaysia.
TC23_2	Validate that filters apply correctly	Map is displayed	1. Open the page. 2. Select 'Dog' from the Animal Type dropdown. 3. Select 'Checked' from the Report	Map updates to show only 'Dog' type animals with 'Checked' report status.

			Status dropdown. 4. Click 'Apply Filters'.	
TC23_3	Validate that search functionality works	Map is displayed	1. Open the page. 2. Enter a valid location in the search input. 3. Click 'Search'.	Map centers on the entered location and zooms in.
TC23_4	Validate that filters reset correctly	Filters are applied	1. Open the page. 2. Apply some filters. 3. Click 'Reset Filters'.	Filters are cleared, and map view returns to default settings.
TC23_5	Validate that markers display correct data	Data is available	1. Open the page. 2. Ensure data is available for markers. 3. Verify that markers are added to the map with correct data.	Markers are displayed with accurate data as per the fetched locations.
TC23_6	Validate that the pie chart updates based on filter selection	Chart and filters are visible	1. Open the page. 2. Select an option from 'animalTypeSelector'. 3. Verify that the pie chart updates accordingly.	Pie chart updates to show selected animal types distribution
TC23_7	Validate that the bar chart updates based on view and data selector	Chart and selectors are visible	1. Open the page. 2. Select 'Yearly' or 'Monthly' view from 'viewSelector'. 3. Select data type	Bar chart updates to reflect the selected view and data type.

			from 'dataSelector'. 4. Verify that the bar chart updates accordingly.	
--	--	--	---	--

6.4.2 Test Data

Table 6.4.2.1 Test Data of User Login Page

Test Data ID	Username	Password
TD1_1	sara	abc12345
TD1_2		
TD1_3	mike	123

Table 6.4.2.2 Test Data of Browse Animals Page

Test Data ID	Species	Age	Search
TD2_1	Dog		
TD2_2		Young	
TD2_3			Dog1
TD2_4			

Table 6.4.2.3 Test Data of Animal Details Page

Test Data ID	Animal ID	Moment ID	Moment Date	Caption
TD3_1	101	201	10/7/2023	Happy animal
TD3_2	102			
TD3_3	103	203	12/8/2023	Playful animal

Table 6.4.2.4 Test Data of Report Stray Animal Page

Test Data ID	Animal Type	Location	Description	Phone Number	Photo
TD4_1	Dog	3.139, 101.686	Brown dog with collar	112312312	dog.jpg
TD4_2	Cat		Small white cat	123456789	cat.jpg
TD4_3		3.139, 101.686	Stray cat near park	198765432	cat2.jpg

Table 6.4.2.5 Test Data of Appointment Information Page

Test Data ID	Appointment ID	Adoption Status	Meeting Location	Submission Date and Time
TD5_1	101	meet_up	Animal Shelter A	13/8/2024 10:00
TD5_2				
TD5_3	102	meet_up	Animal Shelter B	14/8/2024 14:00
TD5_4	103	approved	Animal Shelter C	15/8/2024 9:30

Table 6.4.2.6 Test Data of Update User Details Page

Test Data ID	Username	Old Password	New Password	Confirm Password	Phone	Email	Name	Age
TD6_1	john_doe	oldpass123			123456789	john@example.com	John	30
TD6_2	john_doe		newpass123	newpass123	123456789	john@example.com	John	30
TD6_3	john_doe	oldpass123	newpass123	newpass456	123456789	john@example.com	John	30

Table 6.4.2.7 Test Data of Staff Login Page

Test Data ID	Staff ID	Password
TD7_1	john123	securePass1
TD7_2		
TD7_3	mike456	wrongPass

Table 6.4.2.8 Test Data of Latest Report Page

Test Data ID	Animal Type	Status	Start Date	End Date	Report ID
TD8_1	Dog	Checked	1/8/2024	10/8/2024	R001
TD8_2	Cat	Unchecked	20/7/2024	30/7/2024	R002
TD8_3	Dog	Unchecked	5/8/2024	12/8/2024	R003
TD8_4		Checked			R004
TD8_5	Cat				

Table 6.4.2.9 Test Data of Register New Staff Page

Test Data ID	Staff ID	Password
TD9_1	staff001	strongPass123
TD9_2		
TD9_3	invalidID	weakpass

Table 6.4.2.10 Test Data of Update Staff Details Page

Test Data ID	Staff ID	Old Password	New Password	Confirm Password	Phone	Email	Birthdate
TD10_1	1001	correct_old	newpass123	newpass123	123456789	staff@example.com	1/1/1990
TD10_2	1001		newpass123	newpass123	123456789	staff@example.com	1/1/1990
TD10_3	1001	correct_old	newpass123	newpass321	123456789	staff@example.com	1/1/1990

Table 6.4.2.11 Test Data of View Animal Page

Test Data ID	Animal ID	Name	Species	Breed	Age	Overall Health	Vaccination	Spayed/Neutered	Date of Intake
TD11_1	A001	Max	Dog	Lab	5	Good	Up-to-date	Yes	15/7/2024
TD11_2	A002	Bella	Cat	Siamese	3	Excellent	Up-to-date	No	20/6/2024
TD11_3	A003	Charlie	Dog	Beagle	4	Fair	Needs Vaccinations	Yes	1/8/2024
TD11_4	A004	Daisy	Cat	Persian	2	Poor	Up-to-date	No	

Table 6.4.2.12 Test Data of Animal Profile Update Page

Test Data ID	Name	Species	Breed	Age	Overall Health	Last Grooming Date	Notes	Description	Photo
TD12_1	Max	Dog	Lab	5	Good	15/6/2024	None	Friendly dog	image.jpg
TD12_2	Bella	Cat	Siamese	3	Excellent	1/7/2024	None	Quiet cat	image.jpg

Table 6.4.2.13 Test Data of Add Animal Page

Test Data ID	Photo	Name	Species	Breed	Age	Overall Health	Vaccination	Temperament	Date
TD13_1	photo.jpg	Max	Dog	Beagle	3	Excellent	Up-to-date	Playful	14/8/2024
TD13_2		Max	Dog	Beagle	3	Excellent	Up-to-date	Playful	14/8/2024
TD13_3	photo.jpg	Max		Beagle	3	Excellent	Up-to-date	Playful	14/8/2024
TD13_4	not_an_image.txt	Max	Dog	Beagle	3	Excellent	Up-to-date	Playful	

Table 6.4.2.14 Test Data of Medical Record Page

Test Data ID	Animal ID	Treatment ID
TD14_1	A001	T1001
TD14_2	A002	
TD14_3	A003	T1003
TD14_4	A004	T1004

Table 6.4.2.15 Test Data of Add Treatment Page

Test Data ID	Animal ID	Treatment Date	Next Due Date	Treatment Type	Treatment Name	Veterinarian	Veterinary Clinic	Notes
TD15_1	AN123	15/8/2024	15/9/2024	Preventative Care	Vaccination	Dr. Smith	VetCare Clinic	None
TD15_2	AN124	16/8/2024		Surgical Procedures	Spaying/Neutering	Dr. John	Pet Clinic	N/A
TD15_3	AN125							
TD15_4	AN126	17/8/2024	17/9/2024	Others	Custom Treatment	Dr. Doe	Animal Health Clinic	Mild case

Table 6.4.2.16 Test Data of Update Treatment Page

Test Data ID	Animal ID	Treatment Date	Next Due Date	Treatment Type	Treatment Name	Veterinarian
TD16_1	AN001	13/8/2024	13/9/2024	Medical Treatments	Antibiotics	Dr. Smith
TD16_2	AN002	14/8/2024		Surgical Procedures	Spaying/Neutering	Dr. Doe
TD16_3		15/8/2024		Preventative Care	Vaccination	Dr. Brown

Table 6.4.2.17 Test Data of View Moment Page

Test Data ID	Animal ID	Moment Date	Caption	Moment Photos
TD17_1	A001	1/8/2024	First Moment	photo1.jpg
TD17_2	A002			

TD17_3	A001	1/8/2024	Edited Moment	photo1_edited.jpg
--------	------	----------	------------------	-------------------

Table 6.4.2.18 Test Data of Add Moment Page

Test Data ID	Animal ID	Moment Photo	Moment Date	Caption
TD18_1	12345	valid_photo.jpg	14/8/2024	Cute moment
TD18_2	12345		14/8/2024	Cute moment
TD18_3		valid_photo.jpg	14/8/2024	Cute moment
TD18_4	12345	valid_photo.jpg		Cute moment
TD18_5	12345	valid_photo.jpg	14/8/2024	

Table 6.4.2.19 Test Data of Update Moment Page

Test Data ID	Moment ID	Animal ID	Moment Date	Caption	Photo Upload
TD19_1	MOM001	AN001	1/8/2024	Caption 1	Valid Photo 1
TD19_2	MOM002	AN002	2/8/2024	Caption 2	Valid Photo 2
TD19_3	MOM003	AN003	3/8/2024	Caption 3	Valid Photo 3

Table 6.4.2.20 Test Data of Adoption Page

Test Data ID	Adoption ID	Animal ID	Username	Status	Location	Note
TD20_1	A123	D456	john_doe	adoption_success	Shelter A	Bring documents
TD20_2	A124	C789	jane_doe	application_rejected	Shelter B	Vaccination pending
TD20_3	A125	D101	mike_s	meet_up	Shelter C	Confirm timing

Table 6.4.2.21 Test Data of View Transaction Page

Test Data ID	Start Date	End Date	Payment Method
TD21_1	1/7/2024	15/8/2024	Credit Card
TD21_2	12/7/2024	15/8/2024	Cash
TD21_3	-	-	Online Banking

Table 6.4.2.22 Test Data of Add Transaction Page

Test Data ID	Adoption ID	Animal ID	Username	Date	Description	Amount	Payment Method
TD22_1	A001	ANM001	user123	15/8/24	Adoption Fee	150	Credit Card
TD22_2	A002	ANM002	user456	15/8/2024	Vaccination Fee	75	Online Banking
TD22_3	A003	ANM003	user789	15/8/2024	Neutering Fee	120	E-wallet

Table 6.4.2.23 Test Data of Analysis Page

Test Data ID	Animal Type	Report Status	Location
TD23_1	Dog	Checked	Kuala Lumpur
TD23_2	Cat	Unchecked	Penang
TD23_3	Dog	Checked	Johor Bahru

Table 6.4.2.24 Test Data of Analysis Page Chart Filter

Test Data ID	View	Data Type	Year
TD23_4	Yearly	Reported Stray Animals	
TD23_5	Monthly	Adopted Animals	2023
TD23_6	Yearly	Transaction Amount	

6.4.3 Test Result and Analysis

Table 6.4.3.1 Test Result of User Login

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC1_1	TD1_1	Login successful	Homepage displayed and menu bar added with appointment, update user and logout option.	Pass
TC1_2	TD1_2	Login failed. Display error message "Please fill out this field"	Display error message "Please fill out in this field"	Pass
TC1_3	TD1_3	Login failed. Display error message "Invalid username or Password!"	Display error message "Invalid username or Password!"	Pass

Table 6.4.3.2 Test Result of Browse Animals Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC2_1	TD2_1	The page refreshes showing only animals with species "Dog".	The page displays animals with species "Dog".	Pass
TC2_2	TD2_2	The page refreshes showing only young animals.	The page displays young animals.	Pass
TC2_3	TD2_3	The page refreshes showing animals with names or descriptions containing "Dog1".	The page displays animals with names or descriptions containing "Dog1".	Pass

TC2_4	TD2_4	The page reloads showing all animals, with no filters applied.	The page reloads showing all animals, with no filters applied.	Pass
-------	-------	--	--	------

Table 6.4.3.3 Test Result of Animal Details Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC3_1	TD3_1	Animal details displayed correctly	Details displayed correctly	Pass
TC3_2	TD3_1	Modal appears with date/time fields and submit button	Modal appears as expected	Pass
TC3_3	TD3_1	Modal closes successfully	Modal closes as expected	Pass
TC3_4	TD3_3	Moment records displayed with photos and captions	Moment records displayed correctly	Pass

Table 6.4.3.4 Test Results of Report Stray Animal Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC4_1	TD4_1	Form submitted successfully and data is saved	Data saved, confirmation message shown	Pass
TC4_2	TD4_2	Submission fails. Display error message "Please select a location on the map."	Error message "Please select a location on the map." is displayed	Pass
TC4_3	TD4_3	Submission fails. Display error message "Please select either a dog or a cat."	Error message "Please select either a dog or a cat." is displayed	Pass

Table 6.4.3.5 Test Result of Appointment Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC5_1	TD5_1	The page displays appointment details in a table format.	Appointment details are displayed correctly.	Pass
TC5_2	TD5_2	The page displays the message "No appointments found."	The message "No appointments found." is displayed.	Pass
TC5_3	TD5_3	"Accept Meet Up" and "Reject Meet Up" buttons are displayed.	Buttons are displayed correctly.	Pass
TC5_4	TD5_4	The "Meeting Location" field is displayed.	The "Meeting Location" field is	Pass

			displayed as expected.	
--	--	--	------------------------	--

Table 6.4.3.6 Test Result of Update User Details

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC6_1	TD6_1	Update successful. Show "Update successful!" message	Update successful! Popup displayed	Pass
TC6_2	TD6_2	Update failed. Display error message "Old password is required."	Old password is required.	Pass
TC6_3	TD6_3	Update failed. Display error message "New password and confirm password do not match."	New password and confirm password do not match.	Pass

Table 6.4.3.7 Test Result of Staff Login

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC7_1	TD7_1	Login successful	Homepage displayed with options to view reports, update profile, and logout	Pass
TC7_2	TD7_2	Login failed. Display error message "Please fill out this field"	Display error message "Please fill out this field"	Pass
TC7_3	TD7_3	Login failed. Display error message "Invalid staff ID or Password!"	Display error message "Invalid staff ID or Password!"	Pass

Table 6.4.3.8 Test Result of Latest Report Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC8_1	TD8_1	The report table displays only reports with animal type "Dog".	Only reports with animal type "Dog" are displayed.	Pass
TC8_2	TD8_4	The report table displays only reports with the status "Checked".	Only reports with the status "Checked" are displayed.	Pass
TC8_3	TD8_3	The report table displays only reports within the selected date range.	Only reports within the date range from "2024-08-05" to "2024-08-12" are displayed.	Pass

TC8_4	TD8_4	The status of the report is updated, and a confirmation message "Status updated successfully" appears.	The report status is updated, and a confirmation message "Status updated successfully" appears.	Pass
TC8_5	TD8_5	All filters and search inputs are cleared, and the report table displays all reports without filtering.	All filters and search inputs are cleared, and all reports are displayed.	Pass

Table 6.4.3.9 Test Result of Register New Staff Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC9_1	TD9_1	Registration successful	Redirected to staff dashboard or confirmation page	Pass
TC9_2	TD9_2	Registration failed. Display error message "Please fill out this field"	Display error message "Please fill out this field"	Pass
TC9_3	TD9_3	Registration failed. Display error message "Invalid staff ID or Password!"	Display error message "Invalid staff ID or Password!"	Pass

Table 6.4.3.10 Test Result of Update Staff Details Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC10_1	TD10_1	Details updated successfully. Display success message.	Details updated successfully. Displayed success message.	Pass
TC10_2	TD10_2	Update failed. Display error message "Old password is required."	Update failed. Displayed error message "Old password is required."	Pass
TC10_3	TD10_3	Update failed. Display error message "Passwords do not match."	Update failed. Displayed error message "Passwords do not match."	Pass
TC10_4	TD10_1	Display confirmation dialog.	Displayed confirmation dialog.	Pass
TC10_5	TD10_1	Staff account deleted successfully. Display success message.	Staff account deleted successfully. Displayed success message.	Pass

Table 6.4.3.11 Test Result of View Animal Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC11_1	TD11_1	Relevant animals matching the search term are displayed in the table.	Max and Charlie displayed in the table after search.	Pass
TC11_2	TD11_2	Only animals of the selected species are displayed in the table.	Only Bella displayed in the table after selecting "Cat".	Pass
TC11_3	TD11_3	Only animals with the selected health status are displayed in the table.	Max and Charlie displayed in the table after selecting "Good".	Pass
TC11_4	TD11_4	Only animals with the selected vaccination status are displayed in the table.	Max and Daisy displayed in the table after selecting "Up-to-date".	Pass
TC11_5	TD11_1	Only animals with the selected spayed/neutered status are displayed in the table.	Max and Charlie displayed in the table after selecting "Yes".	Pass
TC11_6	TD11_2	Only animals within the selected date range are displayed in the table.	Max and Daisy displayed in the table after selecting a range that includes their intake dates.	Pass
TC11_7	TD11_3	All filters are cleared and the full list of animals is displayed.	All animals displayed after clicking "Clear".	Pass

TC11_8	TD11_4	User is redirected to the animal profile page.	Successfully redirected to the animal profile page.	Pass
--------	--------	--	---	------

Table 6.4.3.12 Test Result of Animal Profile Update Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC12_1	TD12_1	Profile updated successfully; user is redirected or shown a confirmation message.	Profile updated successfully; confirmation message displayed.	Pass
TC12_2	TD12_2	Submission blocked; error messages for missing fields are displayed.	Error messages displayed for missing fields.	Pass

Table 6.4.3.13 Test Result of Add Animal Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC13_1	TD13_1	All form fields are displayed correctly and ready for input.	All fields displayed correctly.	Pass
TC13_2	TD13_1	Form is submitted successfully, and the user is redirected to the next page.	Form submitted successfully, redirected to the confirmation page.	Pass
TC13_3	TD13_2	Submission fails, and an error message appears indicating missing fields.	Error message displayed: "Please fill out this field."	Pass

TC13_4	TD13_3	Date field is automatically set to the current date.	Date field correctly set to the current date.	Pass
TC13_5	TD13_4	Form submission is prevented, and an error message appears when a non-image file is uploaded to the photo field.	Error message displayed: "Invalid file type. Please upload an image file."	Pass

Table 6.4.3.14 Test Result of Medical Record Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC14_1	TD14_1	The page displays the correct treatment records for the animal_id.	The page displays the correct treatment records.	Pass
TC14_2	TD14_2	The page displays "No treatment records found for this animal."	The page displays "No treatment records found for this animal."	Pass
TC14_3	TD14_3	The page redirects to the new treatment page for the specific animal_id.	The page redirects to the new treatment page.	Pass
TC14_4	TD14_4	The page redirects to the treatment editing page for the specific treatment_id.	The page redirects to the treatment editing page.	Pass

Table 6.4.3.15 Test Result of Add Treatment Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC15_1	TD15_1	Form displays correctly with all fields	Form displayed correctly with all fields	Pass
TC15_2	TD15_1	Treatment details are submitted successfully	Treatment details were submitted successfully	Pass
TC15_3	TD15_2	Error message displayed indicating required fields must be filled	Error message displayed indicating required fields must be filled	Pass
TC15_4	TD15_3	Treatment names are populated correctly based on the treatment type	Treatment names were populated correctly based on the treatment type	Pass
TC15_5	TD15_4	Text input for treatment name appears when 'Others' is selected	Text input for treatment name appeared when 'Others' was selected	Pass

Table 6.4.3.16 Test Result of Update Treatment Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC16_1	TD16_1	Update successful. The treatment record is updated in the database.	Treatment record updated successfully	Pass
TC16_2	TD16_3	Update failed. Display error message "Please fill out this field"	Error message displayed correctly	Pass
TC16_3	TD16_2	Delete successful. The treatment record is removed from the database.	Treatment record deleted successfully	Pass

Table 6.4.3.17 Test Result of View Moment Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC17_1	TD17_1	Moments are displayed with correct photo, date, and caption.	Moments displayed with correct photo, date, and caption.	Pass
TC17_2	TD17_1	Redirect to the "Add Moment" page.	Redirected to the "Add Moment" page.	Pass

TC17_3	TD17_1	Redirect to the "Edit Post" page with the correct moment data pre-filled.	Redirected to the "Edit Post" page with the correct moment data pre-filled.	Pass
TC17_4	TD17_2	Display message "No moments recorded for this animal."	Displayed message "No moments recorded for this animal."	Pass

Table 6.4.3.18 Test Result of Add Moment Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC18_1	TD18_1	Animal ID is displayed automatically and correctly based on the chosen animal.	Animal ID is displayed automatically and correctly based on the chosen animal.	Pass
TC18_2	TD18_2	Form submission fails. Display error message: "Moment photo is required."	Form submission fails. Display error message: "Moment photo is required."	Pass

TC18_3	TD18_3	Form submission fails. Display error message: "Moment date is required."	Form submission fails. Display error message: "Moment date is required."	Pass
TC18_4	TD18_4	Form submission fails. Display error message: "Caption is required."	Form submission fails. Display error message: "Caption is required."	Pass
TC18_5	TD18_5	Form is submitted successfully. User is redirected to a confirmation page or receives a success message.	Form is submitted successfully. User is redirected to a confirmation page or receives a success message.	Pass

Table 6.4.3.19 Test Result of Update Moment Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC19_1	TD19_1	Only the moment with Moment ID MOM001 is displayed.	Only the moment with Moment ID MOM001 is displayed.	Pass
TC19_2	TD19_2	Only the moments with Animal ID AN002 are displayed.	Only the moments with Animal ID AN002 are displayed.	Pass

TC19_3	TD19_3	Only the moments with Moment Date 2024-08-03 are displayed.	Only the moments with Moment Date 2024-08-03 are displayed.	Pass
TC19_4	N/A	All filters are reset, and all moments are visible.	All filters are reset, and all moments are visible.	Pass

Table 6.4.3.20 Test Result of Adoption Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC20_1	TD20_1	Status is updated to "adoption_success" and redirected to payment page.	Status is updated to "adoption_success" and redirected to payment page.	Pass
TC20_2	TD20_2	Location and notes are updated and displayed correctly in the UI	Location and notes are updated and displayed correctly in the UI	Pass
TC20_3	TD20_3	Adoption entry is removed from the table and the database	Adoption entry is removed from the table and the database	Pass

Table 6.4.3.21 Test Result of View Transaction Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC21_1	TD21_1	Transactions are filtered by the selected date range	Transactions are correctly displayed for the date range 2024-08-01 to 2024-08-15	Pass
TC21_2	TD21_2	Transactions are filtered by the selected payment method	Transactions are correctly displayed for "Cash" payment method	Pass
TC21_3	TD21_3	PDF is generated and downloaded with the correct transaction data	PDF is successfully generated with the correct transaction data	Pass

Table 6.4.3.22 Test Result of Add Transaction Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC22_1	TD22_1	All fields are displayed correctly	All fields are displayed as expected	Pass
TC22_2	TD22_2	Date field is auto populated with the current date	Date field is correctly auto populated	Pass
TC22_3	TD22_3	Transaction is successfully added	Transaction is added and user is redirected to the confirmation page	Pass

Table 6.4.3.23 Test Result of Analysis Page

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC23_1		Map is displayed with default view centered on Malaysia.	Map displayed with default view centered on Malaysia.	Pass
TC23_2	TD23_1, TD23_2	Map updates to show only 'Dog' type animals with 'Checked' report status.	Map updated to show only 'Dog' type animals with 'Checked' report status.	Pass
TC23_3		Map centers on the entered location and zooms in.	Map centered on the entered location and zoomed in.	Pass
TC23_4		Filters are cleared, and map view returns to default settings.	Filters cleared, and map view returned to default settings.	Pass
TC23_5	TD23_1, TD23_2, TD23_3	Markers are displayed with accurate data as per the fetched locations.	Markers displayed with accurate data for each location.	Pass
TC23_6	TD23_1, TD23_2	Pie chart updates to show selected animal types distribution.	Pie chart updated to show selected animal types distribution.	Pass
TC23_7	TD23_4, TD23_5	Bar chart updates to reflect the selected view and data type.	Bar chart updated to reflect the selected view and data type.	Pass

6.5 Conclusion

To sum up, “Find Your Pawfect Buddy” system testing is an essential step in the DBLC process. The purpose of the DBLC testing phase is to aid in the process of checking and early defect and error identification. Black and white box tests are conducted in ORPPTS. This chapter completes the test schedule, test description, and test result.



CHAPTER 7: CONCLUSION

7.1 Introduction

This last section provides an overview of the "Find Your Pawfect Buddy" project, evaluating its advantages, disadvantages, suggested enhancements, and impact on the management of stray animals. In addition to outlining the project's impact on animal welfare, the analysis will show how it solves important concerns in stray animal reporting and adoption.

7.2 Observation Weakness and Strengths

7.2.1 Strengths

The strengths of "Find Your Pawfect Buddy" are:

i. Efficiency

The system increases the efficiency of reporting stray animals and expedites the response time of shelters by providing a centralized platform where users can quickly and easily report sightings with specific information.

ii. Adoption Transparency

It provides thorough animal profiles to assist prospective adopters in making educated decisions, thereby raising the probability of adoptions ending in success.

iii. Data Insights

The analytics tool helps shelters plan strategically and allocate resources more effectively by offering insightful information on adoption trends and stray animal trends.

7.2.2 Weaknesses

The weaknesses of “Find Your Pawfect Buddy” are:

i. Recovery Features

There are currently no automatic backup and recovery features on the system, which could lead to data loss in the event of unanticipated problems.

ii. Notification Reminders

The absence of integrated alerts or prompts for personnel and users concerning unfinished business or updates may cause adoption procedures to delay.

iii. Password Management

It could negatively affect user experience and system accessibility if users are unable to retrieve their login credentials. This is because there is no 'forgot password' feature.

7.3 Propositions of improvement

i. Automatic Backup and Recovery

Regular backup and recovery feature implementation will guarantee data availability and integrity in the event of system failures or other problems.

ii. Notification Reminders

It can improve communication and encourage timely actions to add automated notifications and reminders for both users and shelter staff.

iii. Forgot Password Function

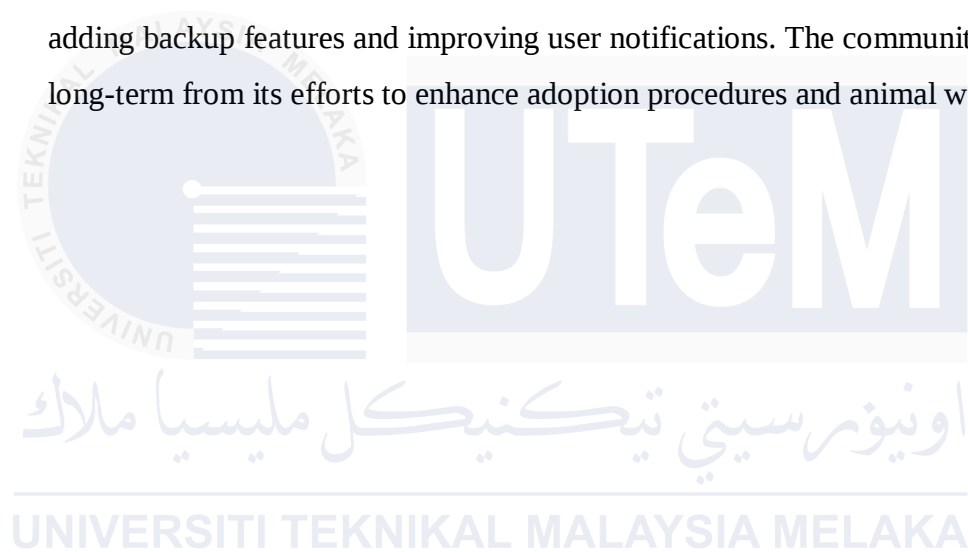
By facilitating easy account recovery, the inclusion of a "forgot password" feature will enhance user accessibility and experience.

7.4 Contribution

The "Find Your Pawfect Buddy" system expedites the process of reporting and adopting stray animals, thereby making a significant contribution to animal welfare. It provides an easy-to-use platform that helps with effective adoption and reporting while also giving useful information for enhancing animal welfare tactics. The system helps create a safer and more peaceful environment for both humans and animals, as well as lessening the amount of stray animals on the streets and raising public awareness of animal adoption.

7.5 Conclusion

The "Find Your Pawfect Buddy" project has successfully improved stray animal management by using a methodical approach, thereby meeting its goals and scope. Because of the structured development process provided by the Waterfall methodology, careful planning, execution, and testing were possible. The project offered a complete solution for reporting, profiling, and adopting stray animals in addition to successfully addressing the inefficiencies in the current systems. Future developments in stray animal management systems will benefit greatly from this project's solid foundation, even though there are still some things to work on, like adding backup features and improving user notifications. The community will benefit long-term from its efforts to enhance adoption procedures and animal welfare.



REFERENCES

Munir, S., Mohd Istajib Mokhtar and Ahmad Firdhaus Arham (2023). Public perspectives on strays and companion animal management in Malaysia. BMC public health, [online] 23(1). doi:<https://doi.org/10.1186/s12889-023-16276-5>.

The ASEAN Post. (2016). Animal Cruelty On The Rise In Malaysia. [online] Available at: <https://theaseanpost.com/article/animal-cruelty-rise-malaysia> [Accessed 20 Jun. 2024].

Paws.org.my. (2023). Adopt – PAWS Malaysia. [online] Available at: <https://www.paws.org.my/adopt/> [Accessed 20 Jun. 2024].

ASPCA. (2015). ASPCA. [online] Available at: <https://www.asPCA.org/> [Accessed 20 Jun. 2024].

