



HAPPY TAILS

BY

MISS RUSSARIN EAIMRITTIKRAI 6488021

MISS SUPITHCHA JONGPHOEMWATTHANAPHON 6488045

MISS KANITA KARUNKITTIKUN 6488049

MISS NISAKORN NGAOSRI 6488226

SECTION 2

ADVISOR

AJ. SNIT SANGLAO

**A Project Submitted in Partial Fulfillment of
the Requirements for
ITCS424 Wireless and Mobile Computing**

Faculty of Information and Communication Technology

Mahidol University

3rd Year Semester 2/2023

COPYRIGHT OF MAHIDOL UNIVERSITY

ACKNOWLEDGEMENTS

We would like to acknowledge the contribution of some people to the completion of this project. We thank our families and colleagues for their cheering support. We have made every effort to carry out this project. We apologize if we have failed to identify some sufficient information. We hope our project will be beneficial to readers.

Miss Russarin Eaimrittikrai

Miss Supithcha Jongphoemwatthanaphon

Miss Kanita Karunkittikun

Miss Nisakorn Ngaosri

HAPPY TAILS

GROUP: PUPPIES

MISS RUSSARIN EAIMRITTIKRAI 6488021 ITCS/B

MISS SUPITHCHA JONGPHOEMWATTHANAPHON 6488045 ITCS/B

MISS KANITA KARUNKITTIKUN 6488049 ITCS/B

MISS NISAKORN NGAOSRI 6488226 ITCS/B

B.Sc. (INFORMATION AND COMMUNICATION TECHNOLOGY)

PROJECT ADVISOR: AJ. SNIT SANGHLAO

ABSTRACT

This project is a part of the Wireless and Mobile Computing course. The objective of this project is to address the challenges that pet owners have been experiencing which is the difficulty in monitoring and recording health records, vaccinations, and medical appointments of their pets. As a result of the observation, there are various mobile applications that individuals utilize on a daily basis to keep track of their health including weight, height, heart rate, and many more. Thus, the team members decide to approach this idea in a similar way for pets to assist and elevate the life qualities of both pet owners and their pets. The scope of this project will cover features for the recording of pets' basic health information, vaccinations, and medical appointments. Moreover, the application will provide some useful tips and advice for new pet owners to gain knowledge about their companions.

CONTENTS

ACKNOWLEDGEMENTS.....	1
ABSTRACT.....	2
LIST OF TABLES.....	6
LIST OF FIGURES.....	7
1 INTRODUCTION.....	9
1.1 Motivation.....	9
1.2 Problem Statement.....	9
1.3 Objectives of the Project.....	10
1.4 Scope of the Project.....	10
1.5 Expected Benefits.....	10
1.6 Organization of the Document.....	10
2 BACKGROUND.....	13
2.1 Literature Review.....	13
3 ANALYSIS AND DESIGN.....	15
3.1 System Architecture Overview.....	15
3.2 System Structure Chart.....	15
3.3 Process Analysis and Design.....	17
3.3.1 Data Flow Diagram.....	17
3.3.2 Data Dictionary.....	18
3.3.2.1 Process Description.....	18
3.3.2.2 Data Stores.....	22
3.3.2.3 Data Element.....	24
3.4 Database Analysis and Design.....	25
3.4.1 ER-Diagram.....	25
3.4.2 Relational Schema.....	26
3.4.3 File Structure.....	28
3.5 I/O Design.....	30

3.5.1	Interface Design.....	30
3.5.2	Transition Diagram.....	36
4	IMPLEMENTATION.....	37
4.1	Hardware and System Environment.....	37
4.2	Implementation Guide and Techniques.....	37
4.2.1	<Guide/Technique/Know-how>.....	37
4.2.2	< Guide/Technique/Know-how>.....	37
5	TESTING AND EVALUATION.....	38
5.1	Unit Tests.....	38
5.1.1	Test Performed on <Process Number> <Process Name>.....	38
5.1.2	Test Performed on <Process Number> <Process Name>.....	38
5.2	System Integration Test.....	38
5.2.1	Test Scenario.....	39
5.2.1.1	39
5.2.1.2	39
5.2.1.3	39
5.2.1.4	39
5.2.1.5	39
6	CONCLUSIONS.....	40
6.1	Benefits.....	40
6.1.1	Benefits to Project Developers.....	40
6.1.2	Benefits to Users.....	40
6.2	Problems and Limitations.....	40
6.3	Future Work.....	40
REFERENCES.....		41
BIOGRAPHIES.....		43

LIST OF TABLES

Table 3.1: List of all Processes.....	17
Table 3.2: Process Description of Register.....	19
Table 3.3: Process Description of Login.....	20
Table 3.4: Process Description of Create a pet profile.....	20
Table 3.5: Process Description of Insert Vaccination and Appointment Records.....	21
Table 3.6: Process Description of Access Vaccination and Appointment Records.....	21
Table 3.7: Process Description of Insert Tips and Tricks.....	21
Table 3.8: Process Description of Access Tips and Tricks.....	22
Table 3.9: Process Description of Access Map.....	22
Table 3.10: List of all Data Stores.....	23
Table 3.11: Data Store Description of User account records.....	23
Table 3.12: Data Store Description of Administrator account records.....	24
Table 3.13: Data Store Description of Pet records.....	24
Table 3.14: Data Store Description of Tips and Trick records.....	24
Table 3.15: Data Store Description of Pet appointment records.....	24
Table 3.16: List of All Data Elements.....	25
Table 3.17: List of all Tables in The System Database.....	27
Table 3.18: File Structure of User.....	28
Table 3.19: File Structure of Administrator.....	29
Table 3.20: File Structure of Pet.....	29
Table 3.21: File Structure of Tips_and_Trick.....	30
Table 3.22: File Structure of Pet_Appointment.....	31
Table 5.1: Test on Process Register.....	51
Table 5.2: Test on Process Login.....	52
Table 5.3: Test on Process Create a Pet Profile.....	53
Table 5.4: Test on Process Insert Vaccination and Appointment Records.....	54

LIST OF FIGURES

Figure 1: Project Timeline.....	10
Figure 2: Purina Pet Health.....	12
Figure 3: System Structure Chart of HAPPY TAILS application.....	15
Figure 3.1: Data Flow Diagram Level 0 of Happy Tails System.....	17
Figure 3.2: Data Flow Diagram Level 1 of Happy Tails System.....	17
Figure 3.3: ER-Diagram of Happy Tails Application.....	27
Figure 3.4: Relational Schema of Happy Tails Application.....	29
Figure 4: Interface Design - Overall of the system.....	34
Figure 5: Interface Design - Account register.....	35
Figure 6: Interface Design - Create Pet Profile.....	35
Figure 7: Interface Design - Vaccination and Appointment Records.....	36
Figure 8: Interface Design - Tips and Tricks.....	37
Figure 9: Interface Design - Map to Nearest Hospitals.....	38
Figure 10: Interface Design - User Profile.....	39
Figure 11: Transition Diagram.....	40
Figure 12: Firebase Cloud Firestore.....	42
Figure 12: Cloud Firebase - User.....	43
Figure 13: Cloud Firebase - Admin.....	44
Figure 14: Cloud Firebase - Pet.....	44
Figure 15: Cloud Firebase - Tips and Tricks.....	45
Figure 16: Cloud Firebase - Pet appointment.....	46
Figure 17: Dart.....	46
Figure 18: Visual Studio Code.....	47
Figure 19: Flowchart of the system - Overview.....	48
Figure 20: Google Maps API.....	50
Figure 21: image_picker.....	51

CHAPTER 1

INTRODUCTION

This chapter serves to introduce readers to the core aspects of the project, including the motivation behind it, the problem statement it seeks to address, the objectives the team hopes to achieve, the scope of the project, the expected benefits, and the organization of the document. These topics have been selected to provide readers with a comprehensive and detailed understanding of the project expecting that this information will provide them with a clear sense of the definite purpose of the project, as well as an appreciation for the thought and care that has gone into its planning and execution.

1.1 Motivation

Members of a family can be in the form of pets which require as much care as human beings. However, it is undeniable that one of the biggest challenges that pet owners have been encountering is keeping track of their pets' health records, vaccinations, and medical appointments which can lead to multiple issues that can negatively impact their pets' lives. Therefore, Happy Tails is a user-friendly mobile application that allows pet owners to manage and monitor their pets' health, including their basic information and medical appointments. Happy Tails aims to comprehensively assist pet owners in monitoring the overall wellness of their pets and provides useful guides for new pet owners who may need supervision due to their lack of experience.

1.2 Problem Statement

1. Most people struggle with tracking their pets' health including vaccinations, medical history, and appointments.
2. Pet owners cannot keep track of their pets' health records leading to missed vaccinations and a lack of timely medical care. Therefore, it can inevitably impact the overall wellness of pets.
3. New pet owners face challenges in understanding and managing their pets' information, which may lead to potential mistakes or oversights that can affect their pets' health.
4. Technologies are not comprehensive enough to manage all aspects of pet health, creating a lack in the care and monitoring process.

1.3 Objectives of the Project

- To facilitate the pet owners in managing and monitoring their pets' health.
- To provide useful guides for new pet owners who may need supervision due to their lack of experience.
- To raise awareness of pet owners to be mindful of their pets' health.
- To improve the pet owners' experience.

1.4 Scope of the Project

Practical mobile application for pet owners to manage and monitor their pets' health such as tracking their pets' health records, vaccinations, medical appointments, etc. In addition, the application can provide useful guides for new pet owners to have more information and assistance in taking care of their pets.

1.5 Expected Benefits

Enabling pet owners worldwide to record and manage health information, vaccinations, and medical appointments of their pets. The application is designed to help pet owners to keep track of and manage their pets. Moreover, the application offers support for those who are new to pet care responsibilities by raising awareness about pet health. It can improve user satisfaction and loyalty, and foster a community of pet owners who trust and depend on the application for the well-being of their pets.

1.6 Organization of the Document

This is the Gantt chart to show the plan to do the project.



Figure 1: Project Timeline

This document consists of 6 chapters including:

1. Introduction: The first chapter is an introduction that explains the overview of the project including motivation, problem statement, object, scope of the project, and expected benefit.
2. Background: The second chapter is a background that covers the topic of the iterative review to provide pertinent information the team has explored and examined.
3. Analysis and Design: The third chapter is an analysis and design of the system that focuses on the processes and system design including system architecture overview, system structure chart, process of analysis and design, data flow diagram, data dictionary, process description, data stores, data element, database analysis and I/O design of the system.
4. Implementation: The fourth chapter is an implementation that demonstrates how the analysis and design aspects discussed in earlier sections are applied in practice to achieve the project's objectives including hardware and system environment and implementation guide and techniques.

5. Testing and Evaluation: The fifth chapter is testing and evaluation of the system that focuses on the methods, results, and implications of testing the system including unit tests and system integration tests.
6. Conclusion: The last chapter is a conclusion that provides a concise overview of the project achievement including the benefits of the project to the developer and user, problems and limitations, and future work of the system.

CHAPTER 2

BACKGROUND

This chapter exclusively delves into the literature review that provides a deeper understanding of the pertinent information the team has explored and examined about this project for readers.

2.1 Literature Review

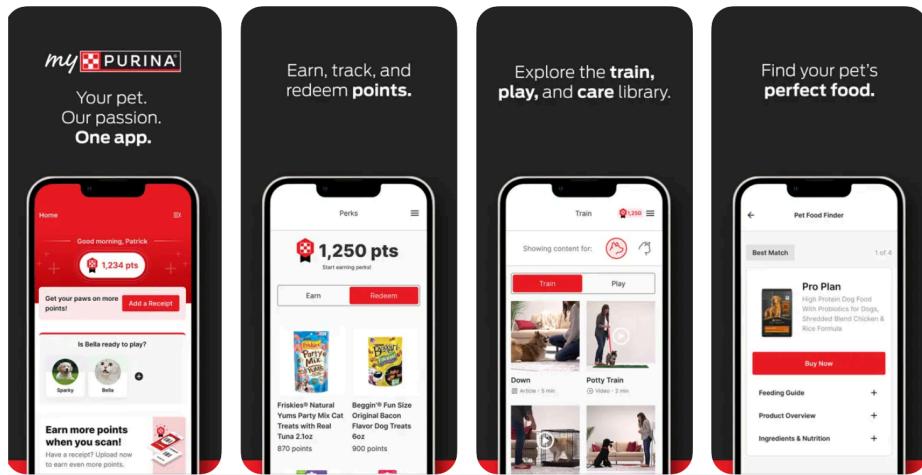


Figure 2: Purina Pet Health

After researching, the team discovered several existing pet care applications such as 'Purina Pet Health'. Purina Pet Health is an application that stores all information related to only dogs and cats via users' smartphones, including vaccination records, veterinary appointment schedules, and grooming appointments to enhance pets' beauty [3]. Moreover, in the case of emergencies, this application can assist in finding nearby animal hospitals or clinics [3]. Nevertheless, the researchers noted its limitations from App Store ratings and reviews, the wide users are experiencing issues in points recording, receipt submission, and unhelpful customer service who are incapable of resolving the issues that customers are experiencing [1]. Even though this application can be downloaded on both IOS and Android platforms, it lacks other language support except English [1]. The team is committed to ensuring that the 'Happy Tails' application is not only informative but also user-friendly to respond to the specific needs and

preferences of pet owners, drawing inspiration from positive aspects observed in similar applications. The team aims to provide a comprehensive and accessible platform that enhances the overall well-being of pets and facilitates between pet owners and applications.

CHAPTER 3

ANALYSIS AND DESIGN

This chapter provides the analysis and design of the application which allows the readers to obtain exclusive information on the system architecture of the application including system structure chart, data flow diagram, and interface design.

3.1 System Architecture Overview

The Happy Tails application will be implemented by the utilization of Flutter. Flutter is an open-source framework developed and produced by Google for creating intuitive, pleasing, natively compiled, multi-platform applications from a single codebase [2]. Flutter comprises a series of independent libraries that each depend on the underlying layer [4]. The layer that developers mostly interact with is Flutter Framework which includes various sets of platforms, layouts, and foundational libraries, composed of a series of layers [4]. These Flutter frameworks incorporate Widgets, App Icons, layout widgets, and other tools and libraries that this project will make use of to create an accessible and user-friendly interface. Furthermore, this project will be using Dart programming language to collaboratively work along with Flutter.

3.2 System Structure Chart

All the features and functionalities of Happy Tails application systems provide including:

- Pets' Health Records: Manages and displays comprehensive health records for each pet such as medical history, conditions, and treatments.
- Vaccination and Appointments Records: Track and remind users about their pets' vaccinations and veterinary appointment schedules.
- Tips and Tricks: Provide essential information and guides for new pet owners, ensuring pet owners have the knowledge needed for proper pet care.
- Map to Nearest Hospitals: Integrates with mapping services to locate and display nearby animal hospitals or clinics in case of emergencies.

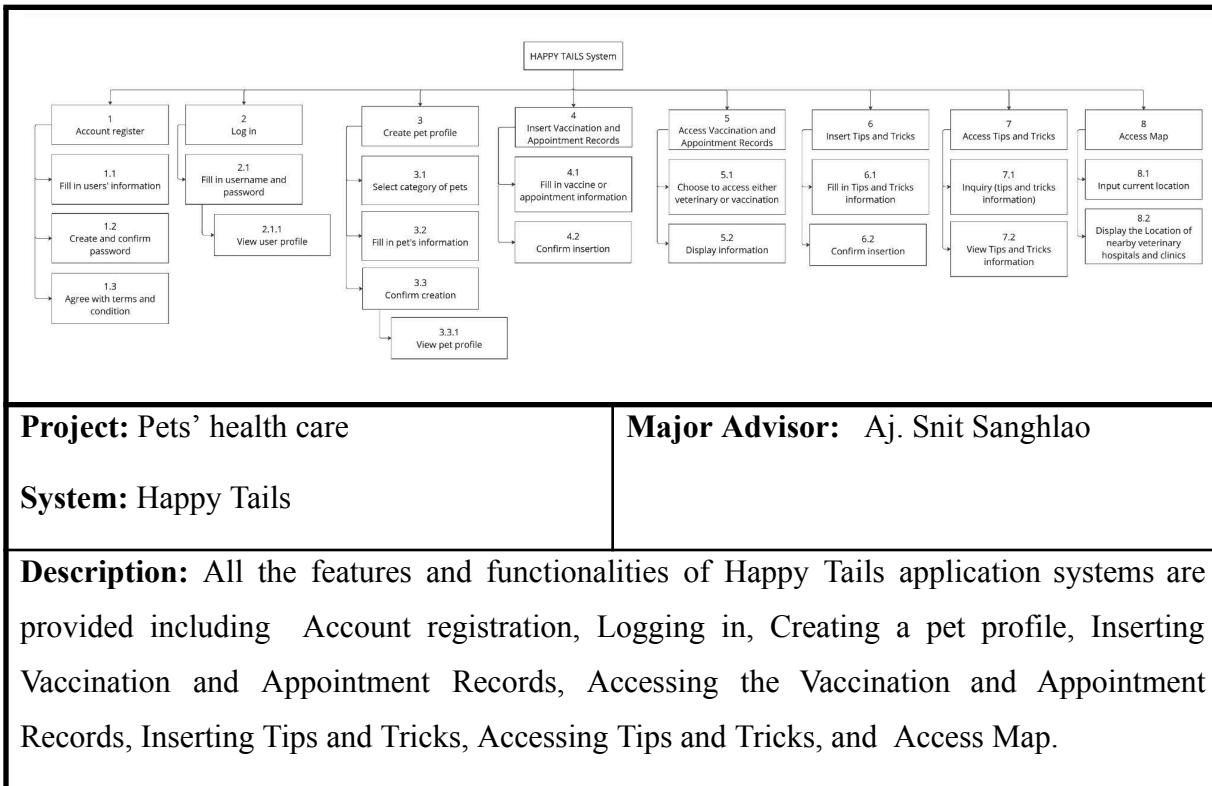


Figure 3: System Structure Chart of HAPPY TAILS application

The detailed description of each subsystem is displayed below:

1. Account register
 - 1.1 Fill in users' information: The user must fill in basic information including full name, and email.
 - 1.2 Create and confirm password: The user must fill in the password and confirm that password.
 - 1.3 Agree with terms and conditions: The user must click an agree terms and conditions button before creating.
2. Log in
 - 2.1 Fill in username and password: The user must fill in the correct username and password to access the system.
 - 2.1.1 View user profile: The user will be able to view his or her profile after logging in.
3. Create a pet profile
 - 3.1 Select category of pets: After successfully logging in, the users will select the category for their pets.

- 3.2 Fill in pet information: After selecting a category, the user must fill in their pet's information.
 - 3.3 Confirm creation: The user must click confirm and the screen will display the creation status confirmation.
 - 3.3.1 View pet profile: The user will be able to view their pet's information.
4. Insert Vaccination and Appointment Records
 - 4.1 Fill in vaccine or appointment information: After successfully logging in, the user can fill in vaccination or veterinary appointment information to create a new record.
 - 4.2 Confirm insertion: The user must click confirm and the screen will display the insertion status confirmation.
5. Access the Vaccination and Appointment Records
 - 5.1 Choose to access either veterinary or vaccination: The user can choose to view either veterinary or vaccination details by clicking the tab.
 - 5.2 Display information: After selecting the tab, the information regarding the veterinary or vaccination will be displayed.
6. Insert Tips and Tricks
 - 6.1 Fill in Tips and Tricks information: the administrator can create a new record of trips and tricks by filling in the information.
 - 6.2 Confirm insertion: The administrator must click confirm and the screen will display the insertion status confirmation.
7. Access Tips and Tricks
 - 7.1 Inquiry (tips and tricks information): the user can search for tips and tricks that they want to read.
 - 7.2 View Tips and Tricks information: After searching, the user can see essential information and guides that they wish to read.
8. Access Map
 - 8.1 Input current location: The user must allow the application to access to their current location.
 - 8.2 Display the Location of nearby veterinary hospitals and clinics: The user can see nearby animal hospitals or clinics in case of emergencies.

3.3 Process Analysis and Design

3.3.1 Data Flow Diagram

These are the data flow diagrams level 0 and 1 which display the stream of data as input and output of entities, process, and data stores.

Project: Pets' health care	Major Advisor: Aj. Snit Sanghlao
System: Happy Tails	
Description: There are 3 major external entities in the system which consist of User, Administrator, and Google Map API.	

Figure 3.1: Data Flow Diagram Level 0 of Happy Tails System

Project: Pets' health care	Major Advisor: Aj. Snit Sanghlao
System: Happy Tails	
Description: There are 3 major external entities in the system which consist of User, Administrator, and Google Map API. Each entity performs processes in the system including Account registration, Logging in, Creating a pet profile, Inserting Vaccination and Appointment Records, Accessing the Vaccination and Appointment Records, Inserting Tips and Tricks, Accessing Tips and Tricks, and Accessing Map.	

Figure 3.2: Data Flow Diagram Level 1 of Happy Tails System

3.3.2 Data Dictionary

A data dictionary is a way to document and describe Processes, Data Stores, and Data Elements (Data Flow) that occur in a Data Flow Diagram (DFD). It is composed of 3 parts as shown below.

- Process Descriptions
- Data Stores
- Data Elements

3.3.2.1 Process Description

This section will provide a detailed description of each process that exists in this system. It includes Inbound Data, Outbound Data, and Logic Summary.

Table 3.1: List of all Processes

No.	Process	Name	Description
1	1	Register	Process for registration.
2	1.1	Fill in users' information	Process to set user name and email for access to the application.
3	1.2	Create and confirm user's password	Process to create and confirm a password.
4	1.3	Agree with terms and conditions	Process to accept agreement of terms of use.
5	2	Login	Process to Login to access the application.
6	2.1	Fill in email and password	Process to enter valid email and password.
7	2.1.1	View user profile	Users can view their profile after successfully login.
7	3	Create a pet profile	Process to create a pet profile the user wants to record their pets' information.

8	3.1	Select categories of pets	Process to select categories of user's pets including dog, cat, hamster, snake, bird, rabbit, fish, and others.
9	3.2	Fill in pet's information	Process to fill in pet's information.
10	3.3	Confirm creation	Process to confirm the creation of pet's profile
11	3.3.1	View pet profile	After that, user can view their pet profile.
12	4	Insert Vaccination and Appointment Records	Process to insert new Vaccination and Veterinary (Appointment) records.
13	4.1	Fill in vaccine or appointment information	Process to fill in Vaccination or Veterinary information including Date of Appointment, Time of Appointment, Type of Appointment, Pet, Location, and Note
14	4.2	Confirm insertion	Process to confirm insertion of Vaccination or Veterinary information.
15	5	Access Vaccination and Appointment Records	Process to access Vaccination and Appointment Records.
16	5.1	Choose to access either the veterinary or vaccination	Process to choose the channels to access Veterinary or Vaccination pages.
17	5.2	Display information	Process to display information based on the user's option.
18	6	Insert Tips and Tricks	Process to insert Tips and Tricks information.
19	6.1	Fill in Tips and Tricks information	Process to fill in Tips and Tricks information for each pet's categories.
20	6.2	Confirm insertion	Process to confirm the insertion of Tips and Tricks.

21	7	Access Tips and Tricks	Process to access Tips and Tricks.
22	7.1	Inquiry (tips and tricks information)	Process to search or inquire Trips and Tricks information.
23	7.2	View Tips and Tricks information	Process to view Trips and Tricks information.
24	8	Access Map	Process to access Map feature.
25	8.1	Input current location	Process to input or enter user's current location.
26	8.2	Display the Location of nearby veterinary hospitals and clinics	Process to display the location of nearby veterinary hospitals and clinics

Table 3.2: Process Description of Register

Process Name	1 - Register
Description	Process for registration.
Inbound data	- User's information
Outbound Data	- Registration validation status
Logic Summary	<pre> graph TD Admin[Administrator account records] --> 2_1[2.1
Fill in username and password] User[User account records] --> 2_1 2_1 -- "Admin's Username (Email) and Password" --> Left 2_1 -- "Login validation status" --> Right 2_1 -- "User's profile" --> Down1 2_1 -- "Admin's profile" --> Down2 </pre>

Table 3.3: Process Description of Login

Process Name	2 - Login
Description	Process to log in to access the application.
Inbound data	<ul style="list-style-type: none"> - Admin's Username (Email) and Password - User's Username (Email) and Password
Outbound Data	<ul style="list-style-type: none"> - Login validation status - User's profile - Admin's profile
Logic Summary	<pre> graph TD A[User account records] --> B[User record] B --> C[2.1
Fill in username and password] D[Admin's Username (Email) and Password] --> C E[User's Username (Email) and Password] --> C C --> F[Login validation status] C --> G[User's profile] C --> H[Admin's profile] </pre>

Table 3.4: Process Description of Create a pet profile

Process Name	3 - Create a pet profile
Description	Process to create a pet profile the user wants to record their pets' information.
Inbound data	<ul style="list-style-type: none"> - Pet's information
Outbound Data	<ul style="list-style-type: none"> - Pet's profile - Confirmation status
Logic Summary	<pre> graph LR A[Pet records] --> B[Pet's information] B --> C[3.1
Select category of pets] C -- "Pet's category" --> D[3.2
Fill in pet's information] D -- "Pet's information" --> E[3.3
Confirm creation] E -- "Pet's profile" --> F[Pet's profile] E -- "Confirmation status" --> G[Confirmation status] E -- Confirmation --> C </pre>

Table 3.5: Process Description of Insert Vaccination and Appointment Records

Process Name	4 - Insert Vaccination and Appointment Records
Description	Process to insert new Vaccination and Veterinary (Appointment) records.
Inbound data	<ul style="list-style-type: none"> - Pet's Vaccination details - Pet's Appointment details - Appointment status
Outbound Data	<ul style="list-style-type: none"> - Confirmation status
Logic Summary	<pre> graph TD A[Pet appointment records] -- "Pet's vaccination and appointment details" --> B[4.1 Fill in vaccine or appointment information] B -- "Vaccination and appointment details" --> C[4.2 Confirm insertion] C -- "Confirmation status" --> D[] B -- "Confirmation" --> C </pre> <p>The logic summary diagram shows a process flow starting from 'Pet appointment records'. An arrow labeled 'Pet's vaccination and appointment details' points to step 4.1. Step 4.1 is labeled 'Fill in vaccine or appointment information'. From step 4.1, an arrow labeled 'Vaccination and appointment details' points to step 4.2. Step 4.2 is labeled 'Confirm insertion'. An arrow labeled 'Confirmation status' points out from step 4.2. Additionally, there is a feedback loop from step 4.1 back to step 4.2 labeled 'Confirmation'.</p>

Table 3.6: Process Description of Access Vaccination and Appointment Records

Process Name	5 - Access Vaccination and Appointment Records
Description	Process to access Vaccination and Appointment Records.
Inbound data	<ul style="list-style-type: none"> - Inquiry(veterinary or vaccination)
Outbound Data	<ul style="list-style-type: none"> - Appointment history - Appointment information
Logic Summary	<pre> graph TD A[Pet appointment records] -- "Pet's vaccination and appointment" --> B[5.1 Choose to access either veterinary or vaccination] B -- "selected option" --> C[5.2 Display information] C -- "Appointment history" --> D[] C -- "Appointment information" --> E[] </pre> <p>The logic summary diagram shows a process flow starting from 'Pet appointment records'. An arrow labeled 'Pet's vaccination and appointment' points to step 5.1. Step 5.1 is labeled 'Choose to access either veterinary or vaccination'. From step 5.1, an arrow labeled 'selected option' points to step 5.2. Step 5.2 is labeled 'Display information'. Two arrows point out from step 5.2: one labeled 'Appointment history' and another labeled 'Appointment information'.</p>

Table 3.7: Process Description of Insert Tips and Tricks

Process Name	6 - Insert Tips and Tricks
Description	Process to insert Tips and Tricks information.

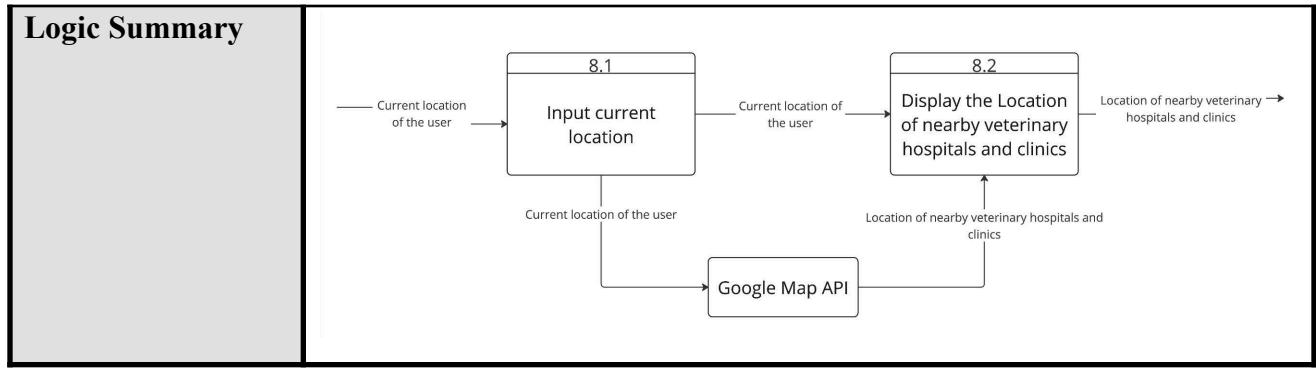
Inbound data	- Tips and Tricks information
Outbound Data	- Confirmation status
Logic Summary	<pre> graph LR A[6.1 Fill in Tips and Tricks information] -- "Tips and Tricks information" --> B[6.2 Confirm insertion] B -- "Confirmation status" --> C[Tips and Trick records] B -- "Tips and Tricks information" --> D[Tips and Trick records] D -- "Confirmation" --> A </pre> <p>The logic summary for process 6 consists of two sequential steps: 6.1 (Fill in Tips and Tricks information) and 6.2 (Confirm insertion). Step 6.1 receives 'Tips and Tricks information' as input and sends 'Confirmation' back to step 6.1 and 'Tips and Tricks information' to step 6.2. Step 6.2 sends 'Confirmation status' as output and 'Tips and Tricks information' to a final output box labeled 'Tips and Trick records'.</p>

Table 3.8: Process Description of Access Tips and Tricks

Process Name	7 - Access Tips and Tricks
Description	Process to access Trips and Tricks.
Inbound data	- Inquiry (tips and tricks information)
Outbound Data	- Tips and Tricks information
Logic Summary	<pre> graph LR A[7.1 Inquiry (tips and tricks information)] -- "Inquiry (tips and tricks information)" --> B[7.2 View Tips and Tricks information] B -- "Tips and Tricks information" --> C[Tips and Trick records] C -- "Inquiry" --> A </pre> <p>The logic summary for process 7 consists of two sequential steps: 7.1 (Inquiry (tips and tricks information)) and 7.2 (View Tips and Tricks information). Step 7.1 receives 'Inquiry (tips and tricks information)' as input and sends 'Inquiry' to step 7.1 and 'Tips and Tricks information' to step 7.2. Step 7.2 sends 'Tips and Tricks information' as output and 'Tips and Trick records' to a final output box.</p>

Table 3.9: Process Description of Access Map

Process Name	8 - Access Map
Description	Process to access Map feature.
Inbound data	- Current location of the user
Outbound Data	- Location of nearby veterinary hospitals and clinics



3.3.2.2 Data Stores

This section describes the data stores that exist in the data flow diagram and consists of the Data Store Name, Description, Inbound Data, and Outbound Data.

Table 3.10: List of all Data Stores

No.	Data Store	Name	Description
1	User account records	User account Database	Collect information of user account including users' information and username and password
2	Administrator account records	Administrator account Database	Collect admins' username and password.
3	Pet Records	Pet Information Database	Collect pets' information.
4	Tips and Trick records	Tips and Trick Database	Collect Tips and Trick information.
5	Pet appointment records	Pet Appointment Database	Collect Appointment (Vaccination and Veterinary) appointment

Table 3.11: Data Store Description of User account records

Data Store Name	User account Database
Description	Collect information of the user account including users' information and username and password.

Inbound data	<ul style="list-style-type: none"> • User's username • User's password • User's information
Outbound Data	<ul style="list-style-type: none"> • User record

Table 3.12: Data Store Description of Administrator account records

Data Store Name	Administrator account Database
Description	Collect admins' username and password.
Inbound data	<ul style="list-style-type: none"> • Admins' username • Admins' password
Outbound Data	<ul style="list-style-type: none"> • Admin record

Table 3.13: Data Store Description of Pet records

Data Store Name	Administrator account Database
Description	Collect admins' username and password.
Inbound data	<ul style="list-style-type: none"> • Pet's information • Pet's vaccination and appointment details
Outbound Data	<ul style="list-style-type: none"> • Pet's vaccination and appointment

Table 3.14: Data Store Description of Tips and Trick records

Data Store Name	Tips and Trick Database
Description	Collect Tips and Trick information.
Inbound data	<ul style="list-style-type: none"> • Inquiry • Tips and Tricks information
Outbound Data	<ul style="list-style-type: none"> • Tips and Tricks information

Table 3.15: Data Store Description of Pet appointment records

Data Store Name	Pet Appointment Database
Description	Collect Appointment (Vaccination and Veterinary) appointment

Inbound data	<ul style="list-style-type: none"> • Pet's vaccination and appointment details
Outbound Data	<ul style="list-style-type: none"> • Pet's vaccination and appointment

3.3.2.3 Data Element

This section describes the data elements or data flows that exist in this system. The table below contains the list of all data elements belonging to their data element name, starting process/source/data store, and ending process/source/data store.

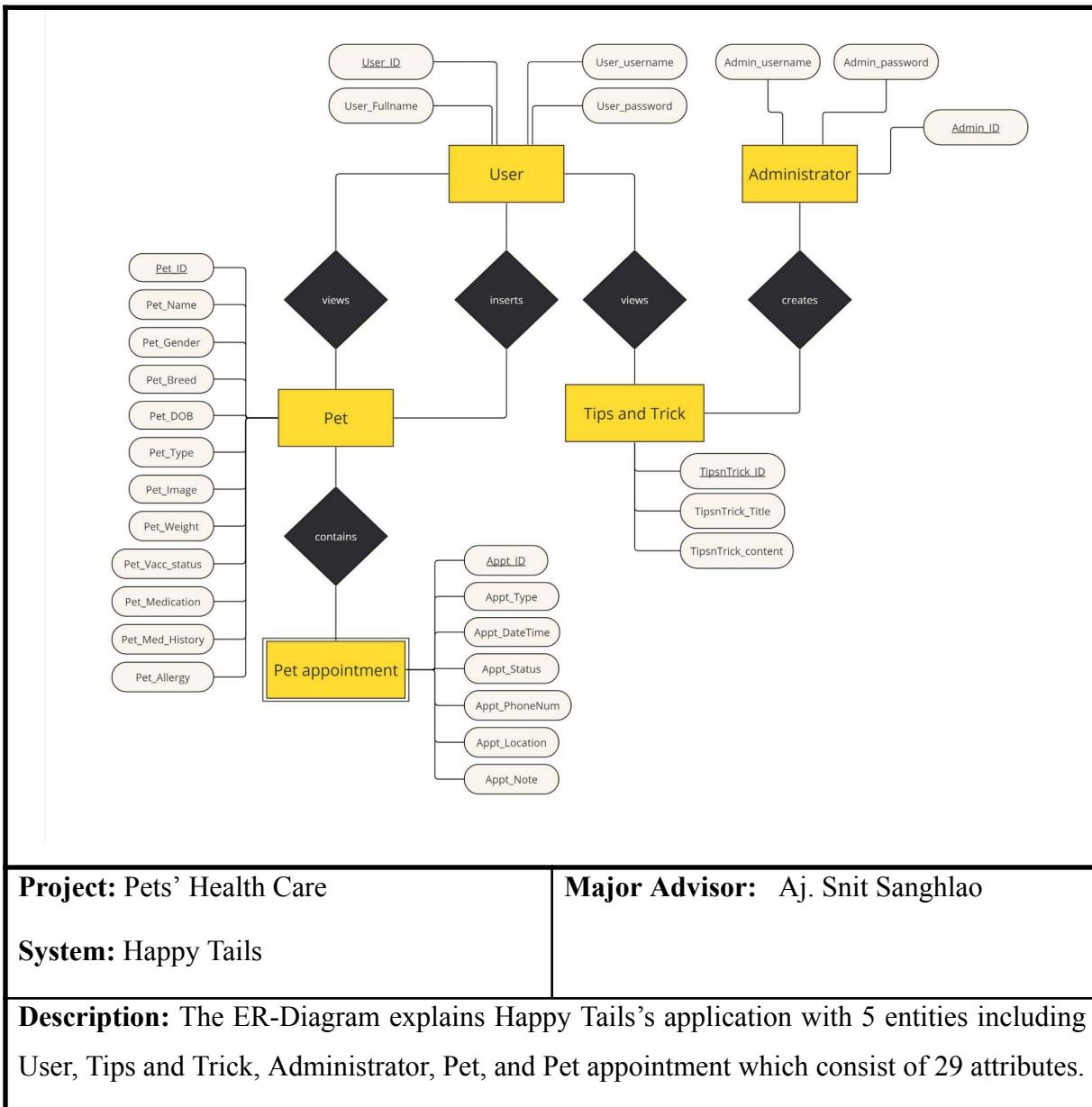
Table 3.16: List of All Data Elements

SEQ	Data Element Name	From Process/Source/Data Store	To Process/Source/Data Store
1	Users' username	User	Registration / Login
2	Users' password	User	Registration / Login
3	User's information	User	Registration
4	Admins' username	Administrator	Login
5	Admins' password	Administrator	Login
6	Pet's information	User	Pet records
7	Pet's vaccination and appointment details	User	Pet records
8	Inquiry	User	Tips and Tricks records
9	Tips and Tricks information	Administrator	Tips and Tricks records
10	Pet's vaccination and appointment details	User	Pet appointment records

3.4 Database Analysis and Design

Database analysis and design, including the ER-Diagram, Relational Diagram, and File Structure, will be covered in this topic to provide readers comprehensive information regarding data within this system.

3.4.1 ER-Diagram



Project: Pets' Health Care

System: Happy Tails

Major Advisor: Aj. Snit Sanghlao

Description: The ER-Diagram explains Happy Tails's application with 5 entities including User, Tips and Trick, Administrator, Pet, and Pet appointment which consist of 29 attributes.

Figure 3.3: ER-Diagram of Happy Tails Application

3.4.2 Relational Schema

This section describes the attributes of the tables in the database. The attribute notation is shown below.

- **Attributes** – which are bold and underlined are the Primary Keys
- *Attributes* – which are Italics are the Foreign Keys
- **Attributes** – which are bold, italic, and underlined are both Primary Keys and Foreign Keys

Tables in this system can be divided into 3 groups as follows:

- Master File Table
- Base File Table
- Transaction File Table

Table 3.17: List of all Tables in The System Database

Table#	Table Name	Table Type	Description
1	User	Master	Stores user's information
2	Administrator	Master	Stores Administrator's information
3	Pet	Base	Stores pet's information
4	Pet_Appointment	Transaction	Stores pet appointment's information
5	Tips_and_Trick	Base	Store tips and tricks information

1. Relational Schema of Master File Tables
 - User
 - Administrator
2. Relational Schema of Base File Tables
 - Pet
 - Tips_and_Trick
3. Relational Schema of Transaction File Tables
 - Pet_Appointment



Figure 3.4: Relational Schema of Happy Tails Application

3.4.3 File Structure

This section shows the details of each file component including field name, field description, field data type, field length, null value, primary key, and foreign key.

Table 3.18: File Structure of User

Table Name: User Table# 01						
Table Type: Master						
Description: This is the file structure of the User that collects the user's information.						
Field Name	Type	Length	Description	Key	Reference	Null
User_ID	Integer	5 Bytes	Store the user's ID.	PK	-	-
User_Fullname	Text string	5 Bytes	Store the user's full name.	-	-	-
User_Username	Text string	5 Bytes	Store the user's username.	-	-	-
User_Password	Text string	5 Bytes	Store the user's password.	-	-	-
User_Img	Bytes	10 Bytes	Store the user's image.			

Total	30	Bytes
-------	----	-------

Table 3.19: File Structure of Administrator

Table Name: Administrator							Table# 02
Table Type: Master							
Description: This is the file structure of the ‘Administrator’ that collects the administrator’s information.							
Field Name	Type	Length	Description	Key	Reference	Null	
Admin_ID	Integer	5 Bytes	Store the ID of each administrator.	PK	-	-	
Admin_username	Text string	5 Bytes	Store the admin’s username.	-	-	-	
Admin_password	Text string	5 Bytes	Store the admin’s password.	-	-	-	
Total	15	Bytes					

Table 3.20: File Structure of Pet

Table Name: Pet							Table# 03
Table Type: Base							
Description: This is the file structure of ‘Pet’ that collects pet’s information.							
Field Name	Type	Length	Description	Key	Reference	Null	
Pet_ID	Integer	5 Bytes	Store the ID of each pet.	PK	-	-	
User_ID	Text string	5 Bytes	The foreign key of a pet is User_ID.	FK	-	-	
Pet_Name	Text string	5 Bytes	Store the pet’s name.	-	-	-	

Pet_Gender	Text string	5 Bytes	Store the pet's gender.	-	-	-
Pet_Breed	Bytes	5 Bytes	Store the pet's breed	-	-	-
Pet_DOB	Date and time	5 Bytes	Store the pet's date of birth.	-	-	-
Pet_Type	Text string	5 Bytes	Store the pet's type.	-	-	-
Pet_Image	Bytes	10 Bytes	Store the pet's image.	-	-	-
Pet_Weight	Floating-point number	5 Bytes	Store the pet's weight.	-	-	-
Pet_Vacc_status	Text string	5 Bytes	Store the pet's vaccination status.	-	-	-
Pet_Med	Text string	5 Bytes	Store the pet's medication.	-	-	-
Pet_Med_History	Text string	5 Bytes	Store the pet's medication history.	-	-	-
Pet_Allergy	Text string	5 Bytes	Store the pet's allergies.	-	-	-
Total	70	Bytes				

Table 3.21: File Structure of Tips_and_Trick

Table Name: Tips_and_Trick		Table# 04				
Table Type:	Base					
Description:	This is the file structure of 'Tips_and_Trick' that collects information on tips and trick					
Field Name	Type	Length	Description	Key	Reference	Null

<u>TipsnTrick_ID</u>	Integer	5 Bytes	Store the ID of tip and trick.	PK	-	-	
<u>Admin_ID</u>	Integer	5 Bytes	The foreign key of a tip and trick is Admin_ID.	FK	Tips_and_Track	-	
<u>TipsnTrick_Title</u>	Text string	5 Bytes	Store the title of a tip and trick.	-	-	-	
<u>TipsnTrick_content</u>	Text string	5 Bytes	Store the tip and trick's content.	-	-	-	
Total	20	Bytes					

Table 3.22: File Structure of Pet_Appointment

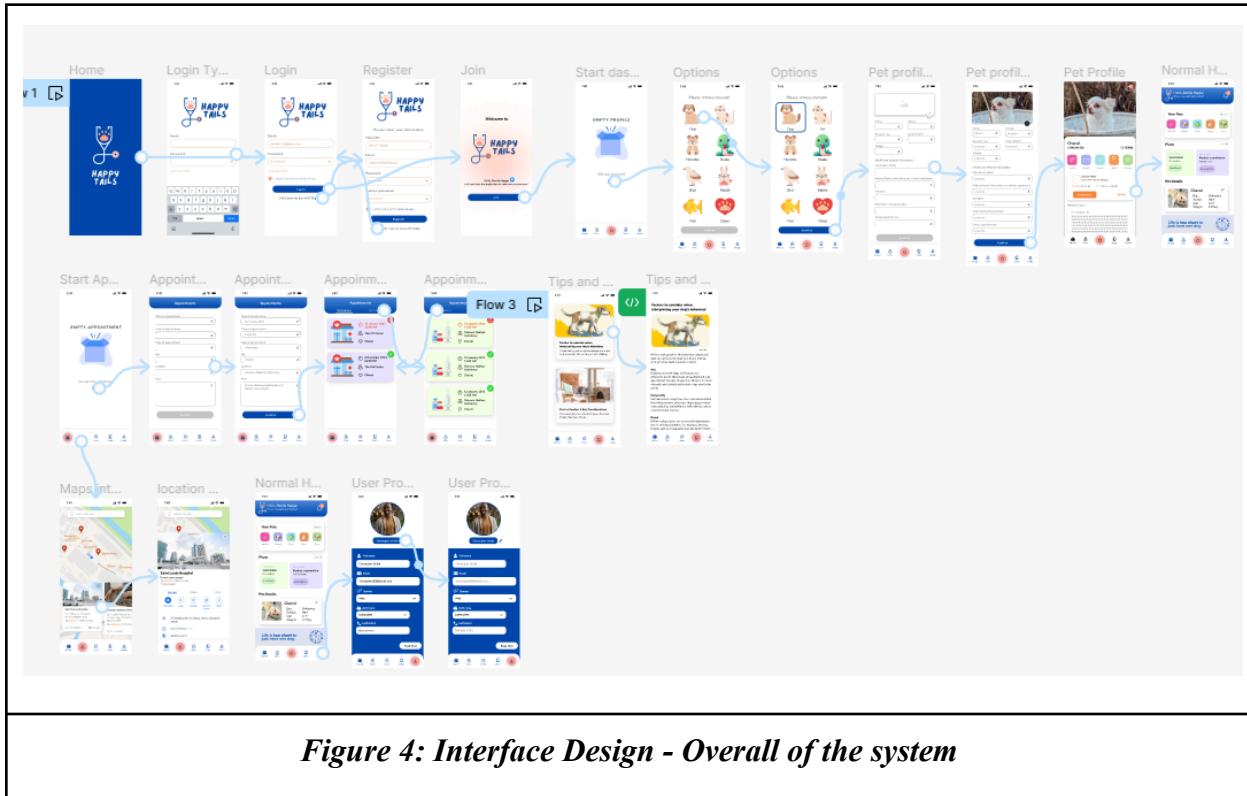
Table Name: Pet_Appointment							Table# 05
Table Type: Transaction							
Field Name	Type	Length	Description	Key	Reference	Null	
<u>Appt_ID</u>	Integer	5 Bytes	Store the ID of the appointment.	PK	-	-	
<u>Pet_ID</u>	Text string	5 Bytes	Store the ID of pets that own this appointment.	FK	-	-	
Appt_Type	Text string	5 Bytes	Store the types of appointments.	-	-	-	
Appt_DateTime	Date and time	5 Bytes	Store the date and time of each appointment.	-	-	-	
Appt_Status	Bytes	5 Bytes	Store the appointment status.	-	-	-	

Appt_PhoneNum	Integer	5 Bytes	Store the contact phone number of each appointment.	-	-	-
Appt_Location	Map	5 Bytes	Store the location of each appointment.	-	-	-
Appt_Note	Text string	5 Bytes	Store the note of each appointment.	-	-	-
Total	40	Bytes				

3.5 I/O Design

This section explains the design of the Input and Output User Interface. The section consists of two parts, the interface design and the transition diagram showing the transition through the system.

3.5.1 Interface Design



The interface design for the Happy Tails application was crafted using the Figma tool. The interface design aims to feature a user-friendly layout with several pages to cater to various functionalities. This figure displays the overall flow of the Happy Tails application's interface. The first row shows the flow of user login, registration, and creating a new pet profile. In addition, the second row shows the flow of creating new vaccinations or veterinary appointments and trips and tricks. Finally, the last row shows the flow of accessing the map to the nearest hospital and user profile page.

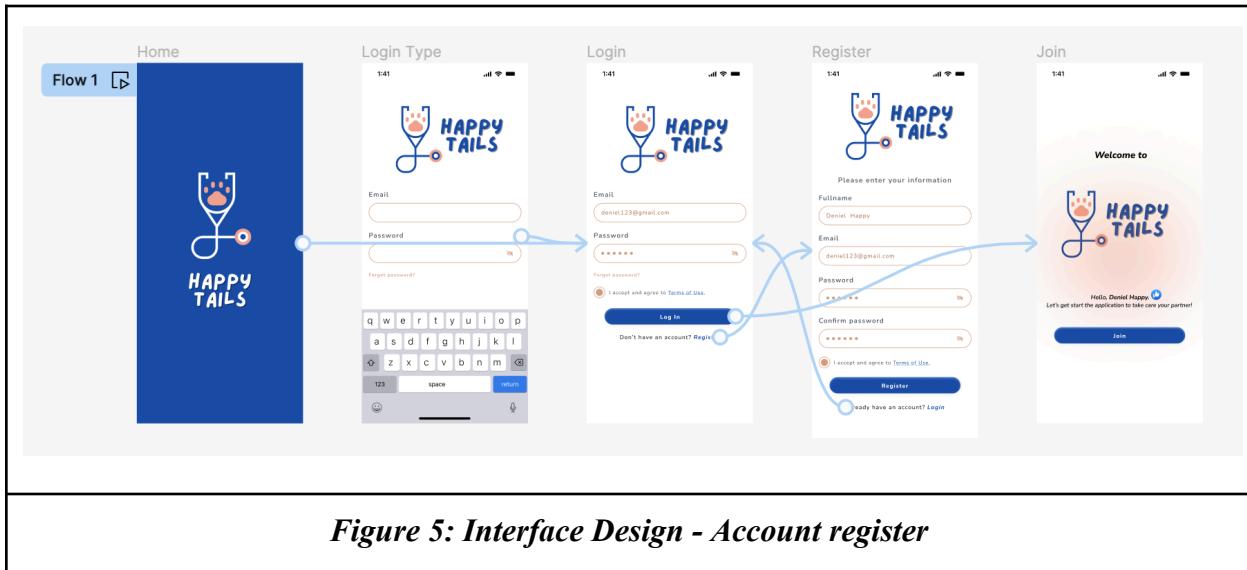


Figure 5: Interface Design - Account register

The account register system flow shows the process when users begin to access the Happy Tails application. The users are required to fill in basic information including full name, email, and password, and confirm a password for creating the user's account. After that, the systems will allow the users to log in through email and a valid password, and access the Happy Tails application.

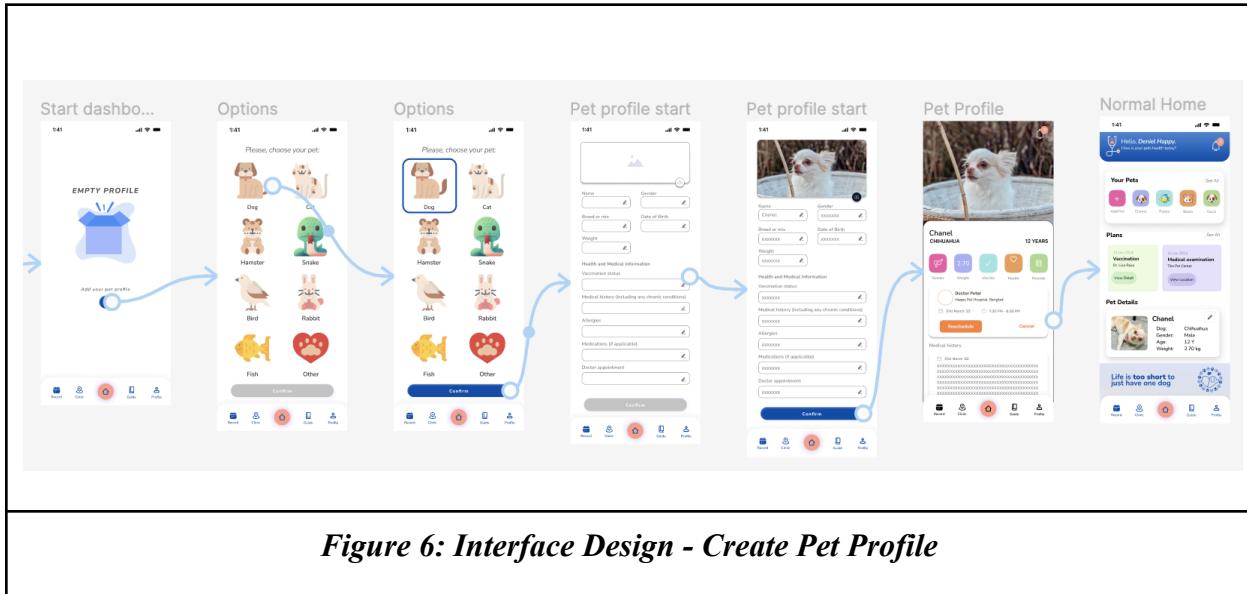


Figure 6: Interface Design - Create Pet Profile

The Create Pet Profile system flow displays the process of creating a pet profile in the Happy Tails application. To create a new pet profile, the users can click on the plus button on the home page which leads to the next page for selecting the type of pet. Once the users select the type of pet, the next page will display the text boxes for receiving user input about the pet's

information. After the users complete inputting details and click confirm, the next page will display the information of the pet profile as can be seen on the last page on the diagram.

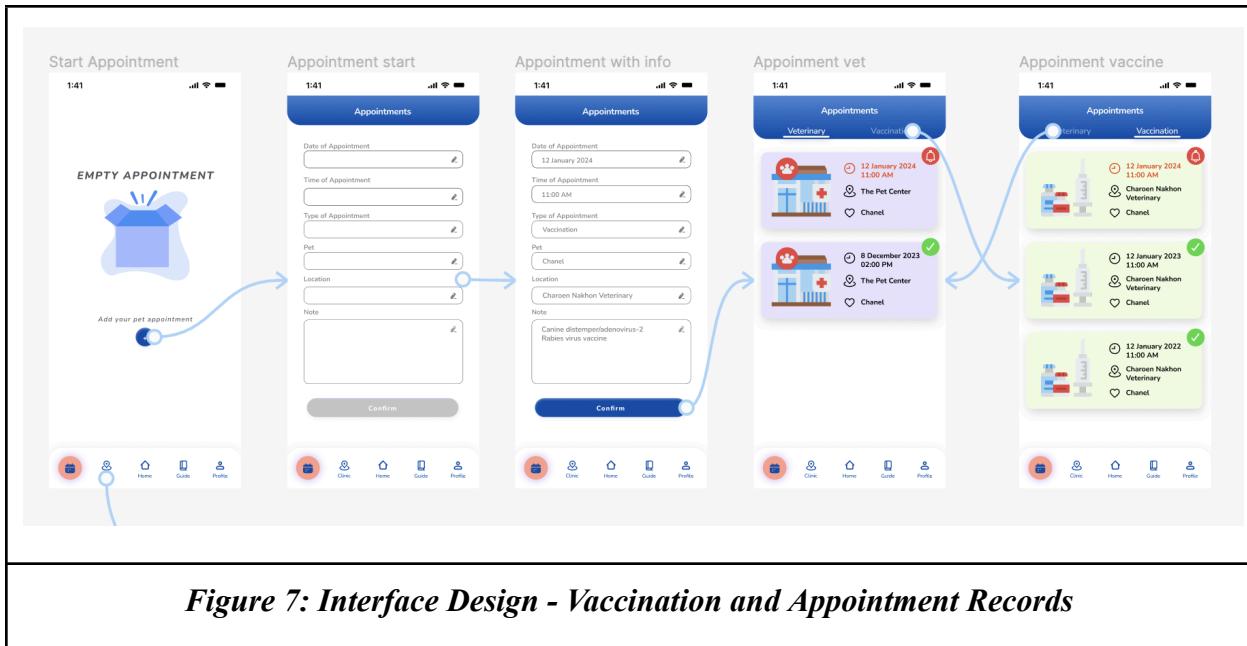


Figure 7: Interface Design - Vaccination and Appointment Records

The vaccination and appointment records system flow displays the process of recording a pet's appointment information. Initially, the users can click on the plus button on the appointment page which leads to the next page displaying the text boxes for receiving user input about the pet's appointment. The users can fill in the date, time, type of appointment, pet name, location, and a short note. After the users complete inputting details and click confirm, the next page will display all medical appointments and vaccination history that can be seen on the last 2 pages on the figure.

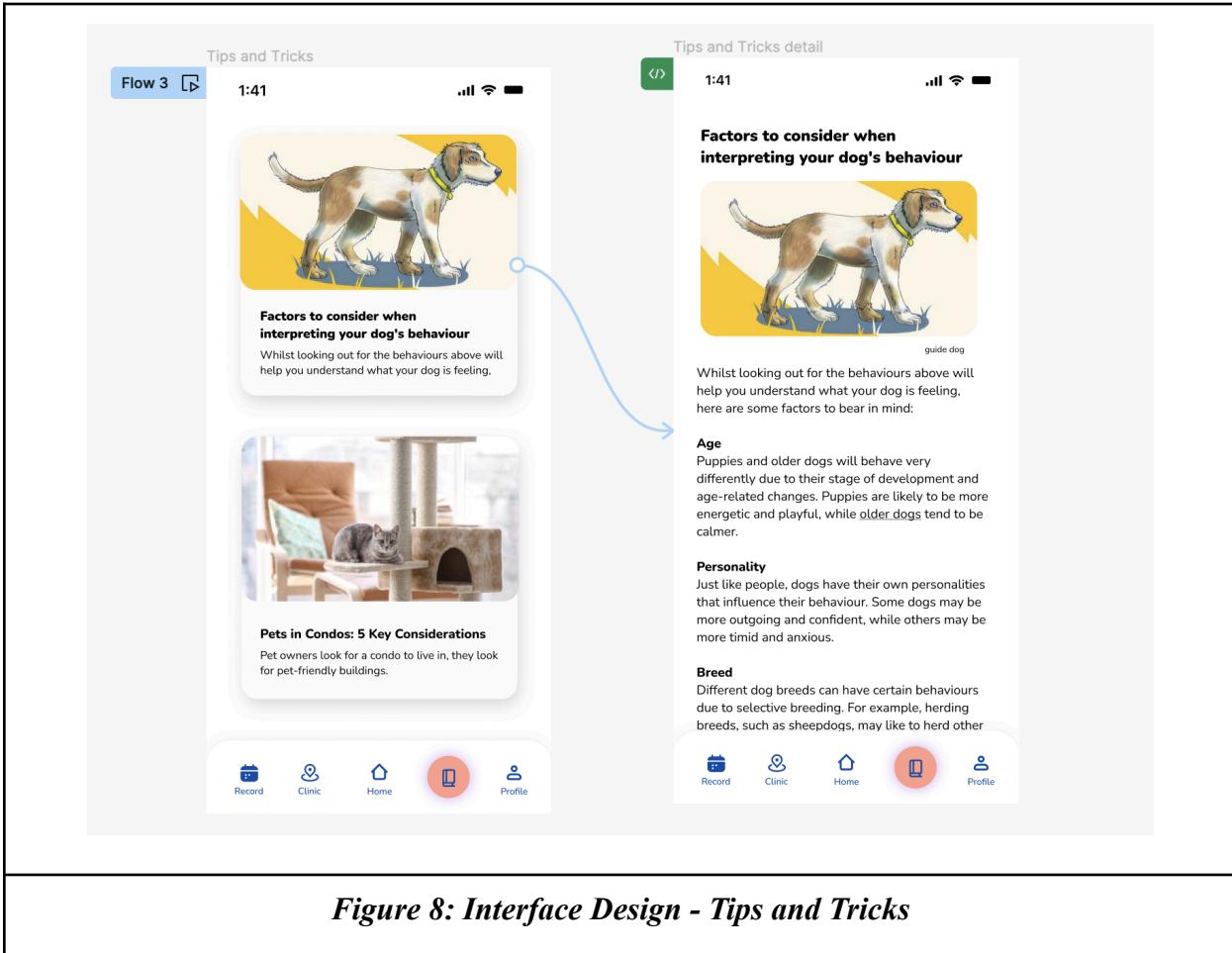


Figure 8: Interface Design - Tips and Tricks

The tips and tricks system flow displays the example of the interface when users visit this page to obtain advice and information about interesting pets or details that aid owner pets in understanding their pets better. The application displays this information in the form of flashcards, each displaying a specific topic. If the users encounter a topic that they are interested in, they can click on the flashcard to read more as can be seen on the second page on the figure.

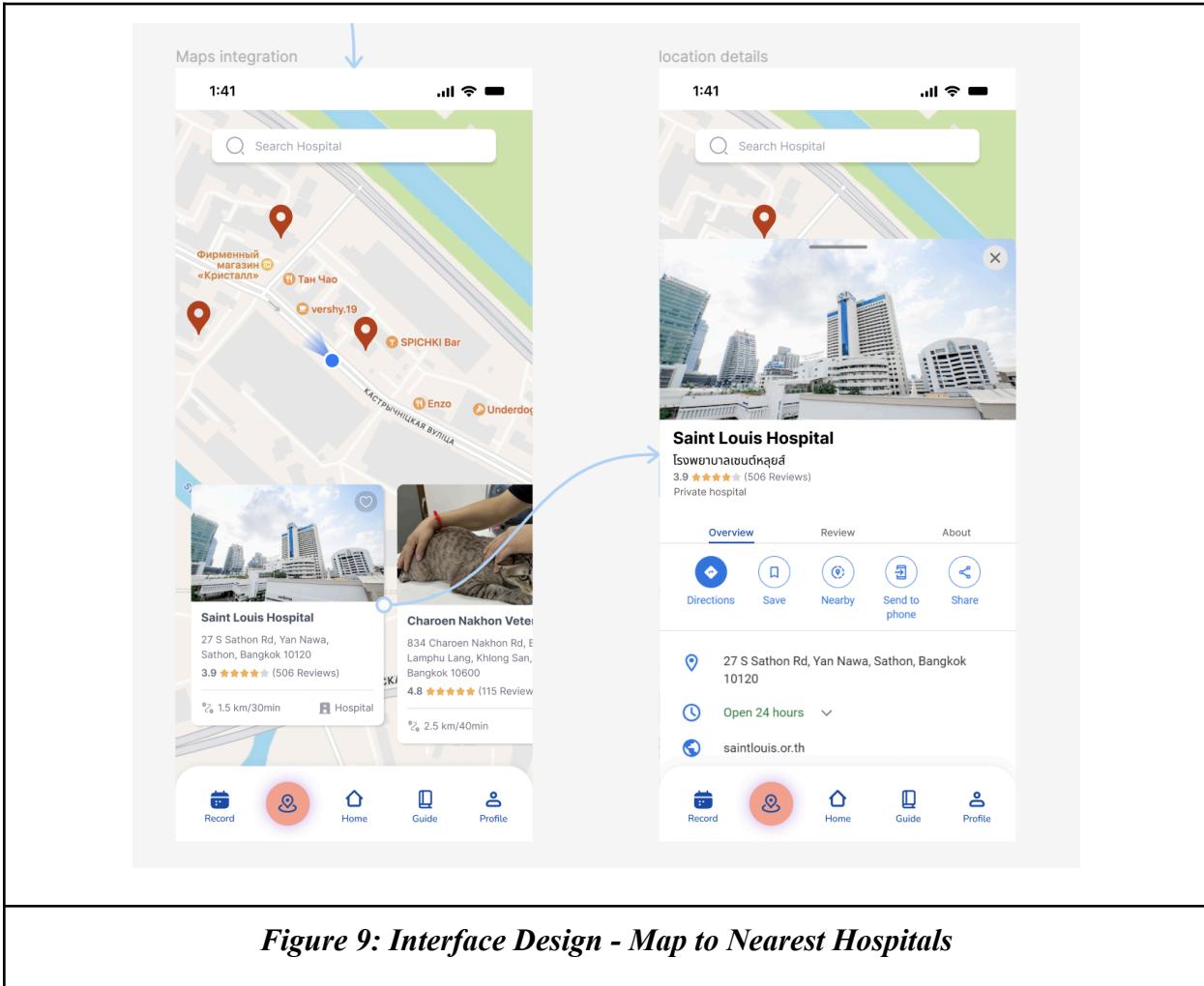


Figure 9: Interface Design - Map to Nearest Hospitals

The Map to Nearest Hospitals system flow is designed to swiftly locate and display nearby hospitals or clinics around the users' location in case of emergency within the Happy Tails application. When the users click on a specific hospital or clinic icon on the map, a pop-up card promptly appears, presenting comprehensive details about that location. This includes vital information such as the facility's name, precise address, contact details, and any other pertinent information relevant to the user's needs during critical situations.

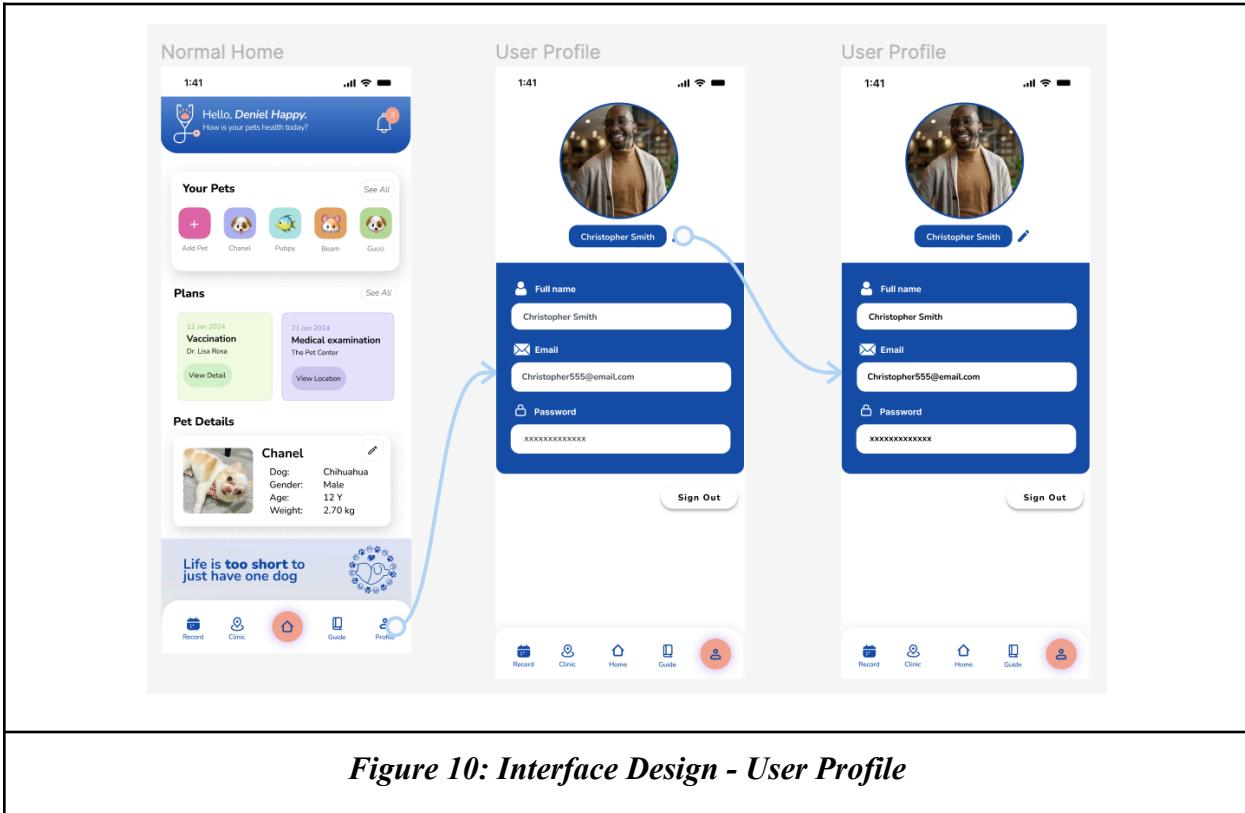
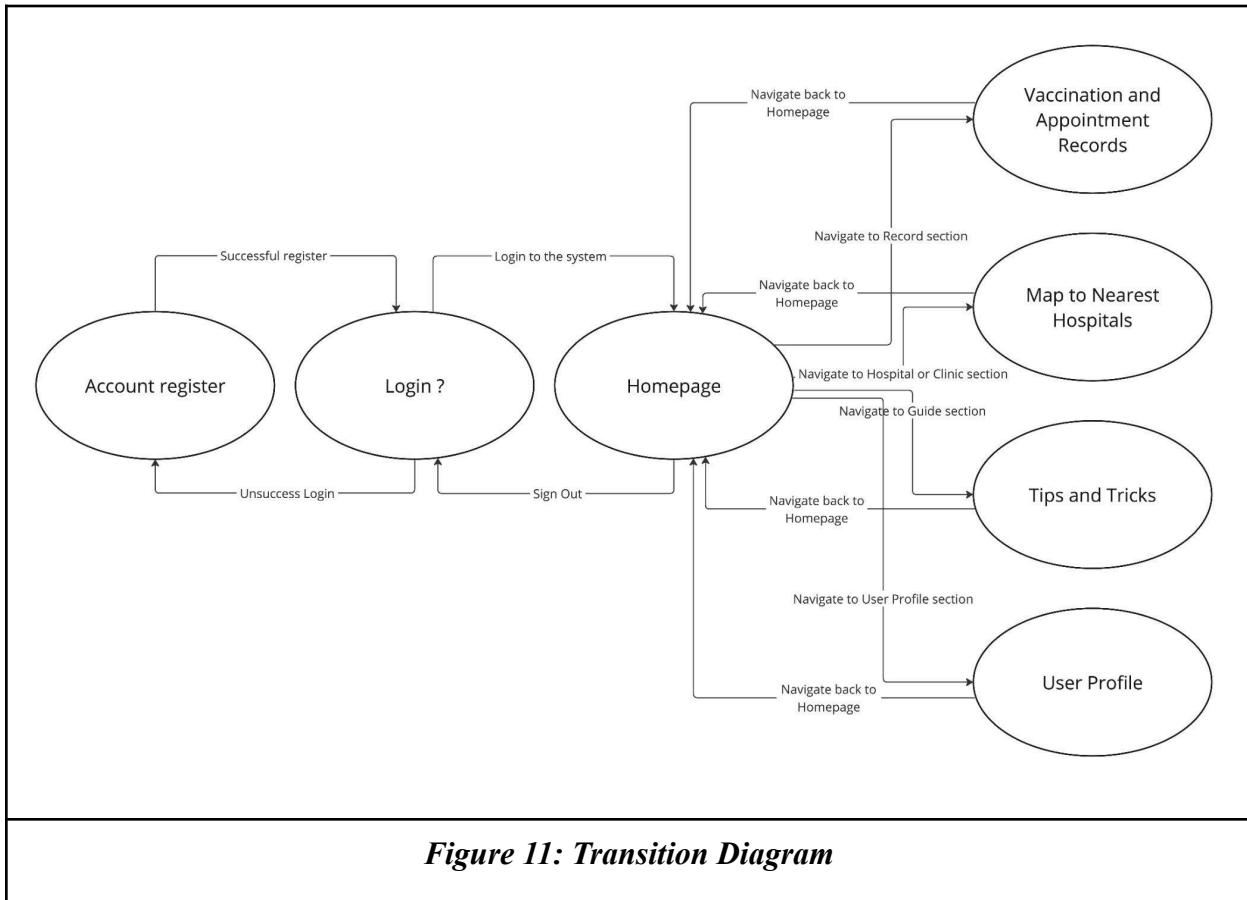


Figure 10: Interface Design - User Profile

The User Profile system flow is designed to allow the user to modify their information including full name, email, and password. The user will also be able to sign out of the application on this page.

3.5.2 Transition Diagram



The transition diagram of the Happy Tails application visualizes the flow of user interactions within the application. Initially, the user begins with the account registration process, where they provide their information. Upon successful registration, the flow moves to the login page. After logging in, the user is directed to the application's homepage, where they can access various features through the navigation bar.

The homepage section serves as a central hub to access every section in the application through the navigation bar. If the users desire to see the vaccination and appointment records, they can access this section from the navigation bar that allows users to add their pet's veterinary and vaccination details, set reminders for upcoming appointments, and track past appointments. The Map to Nearest Hospitals feature provides users with the locations of nearby pet hospitals or clinics based on their current location, particularly useful in emergencies. The Tips and Tricks section, also accessible from the navigation bar, provides valuable information on pet care. Lastly, users can access their profile or sign out using the user profile navigation.

CHAPTER 4

IMPLEMENTATION

4.1 Hardware and System Environment

- Operating System and Utilities Applications
 - o Hardware
 - Laptop hardware 1: macOS Sonoma Version 14.2.1, 2 GHz Quad-Core Intel Core i5, RAM 16 GB
 - Laptop hardware 2: MSI 13th Gen Intel(R) Core(TM) i7-13620H 2.40 GHz, RAM 16 GB
 - Laptop hardware 3: macOS Sonoma Version 14.2.1, 2.3 GHz Quad-Core Intel Core i5, RAM 8 GB
 - Laptop hardware 4: HP Laptop 15s-eq2xxx Ryzen 7 5700U, 1.80 GHz, RAM 16 GB
 - o Software
 - Operation system: Windows 11 Pro version 23H2, macOS Sonoma Version 14.2.1
 - Visual Studio, Google Chrome, Safari, Google Document, Firebase, Flutter
- Web Server Software
 - o Google Cloud and Firebase

Firebase is a collection of cloud-based development tools operated by Google to assist developers in deploying, building, and scaling their website and mobile applications [6]. It provides a wide range of features such as Authentication that can support email and password login, Google Sign-In, Facebook Login, and more; Performance Monitoring which provides insights into the performance of the users' applications and tracks additional metrics like CPU usage, memory usage, and network traffic; Realtime Database which is a cloud-hosted NoSQL database allowing the users to store and retrieve in real-time across all users' devices [6].
- Database Management System (DBMS)



Cloud
Firestore

Figure 12: Firebase Cloud Firestore

- o Firebase Cloud Firestore:

This project leveraged the Firebase Cloud Firestore to manipulate and connect between the application and the database. Cloud Firestore is a cloud-hosted that is built on Google Cloud infrastructure. It facilitates storage and synchronization of data that can be accessed via native SDKs through several devices such as Apple, Android, and web apps. Besides enabling real-time sync across client apps, it also supports offline functionality for mobile and web with seamless integration with other Firebase and Google Cloud products [7]. Moreover, querying in Cloud Firestore is significantly efficient and flexible since the process of retrieving data can be done without the need to fetch the entire collection or any nested subcollections [7].

The screenshot shows the Firebase Cloud Firestore interface. The path is User > 914AwxOpuFxD.. The main view displays a document named "User" with fields: User_Fullname ("Danny Meyer"), User_ID (1), User_Img (""), User_password ("dmeyer123"), and User_username ("danny_m"). The left sidebar shows collections: (default), Admin, Pet, Tips and Trick, and User (selected). Under the User collection, a subcollection named "914AwxOpuFxDcEcmbNjb" is expanded, showing documents: 14wh1APgD0Cq4FKuDpbL and E0ko9E8s05H5KEAjMsdM.

The screenshot shows the Cloud Firestore interface with two separate document snapshots displayed side-by-side.

Document 1: Alice Johnson

- Collection:** User
- Document ID:** YWb4LJAizhR0JihvgtOj
- Fields:**
 - User_Fullname: "Alice Johnson"
 - User_ID: 2
 - User_Img: ""
 - User_password: "passw0rd123"
 - User_username: "alice_johnson"

Document 2: Bob Smith

- Collection:** User
- Document ID:** 14wH1APgDOCq4FKuDpbL
- Fields:**
 - User_Fullname: "Bob Smith"
 - User_ID: 3
 - User_Img: ""
 - User_password: "securePass456"
 - User_username: "bob_smith"

Figure 12: Cloud Firebase - User

The screenshot shows the Cloud Firestore interface with one document snapshot displayed.

Collection: Admin

Document ID: ZJCRmVqUDjn9.

Fields:

- Admin_ID: 1
- Admin_password: "Adminpw1"
- Admin_username: "Admin01"

The screenshot shows the Cloud Firestore Admin interface. The left sidebar lists collections: (default), Admin, Pet, Tips and Trick, and User. The Admin collection is selected. The main area shows a document named "Admin" with fields: Admin_ID: 2, Admin_password: "Adminpw2", and Admin_username: "Admin2".

Figure 13: Cloud Firebase - Admin

The screenshot shows the Cloud Firestore Admin interface. The left sidebar lists collections: (default), Pet, and User. The Pet collection is selected. A document named "zjH3yzj809pQq..." is shown with fields: Pet_Allergy: "Salmon", Pet_Breed: "Chihuahua", Pet_DOB: "3 June 2013 at 22:22:22 UTC+7", Pet_Gender: "Male", Pet_ID: 1, Pet_Image: "Chanel.png", Pet_Med_History: "last week", Pet_Medication: "penicillin, trimethoprim-sulfa, cephalexin and enrofloxacin", Pet_Name: "Chanel", Pet_Type: "Dog", Pet_Vacc_Status: "Pending", and Pet_Weight: "15 Kg".

Figure 14: Cloud Firebase - Pet

(default)		Tips and Trick	More in Google Cloud
+ Start collection Admin Pet Tips and Trick > User		+ Add document APiEcMLBQKmPgHr5kenv xhbpBk11568rLJqKv5gS >	+ Start collection + Add field TipsnTrick_ID: "1" TipsnTrick_Title: "Factor to consider when interpreting your dog's behaviour" TipsnTrick_Content: "Whilst looking out for the behaviors above will help you understand what your dog is feeling, here are some factors to bear in mind: It's essential to consider the specific situation and what's normal for your dog when reading their body language. What might be relaxed behavior in a home environment can mean something different in a public setting. Puppies and older dogs will behave very differently due to their stage of development and age-related changes. Puppies are likely to be more energetic and playful, while older dogs tend to be calmer. Just like people, dogs have their own personalities that influence their behavior. Some dogs may be more outgoing."
(default)		Tips and Trick	More in Google Cloud
+ Start collection Admin Pet Tips and Trick > User		+ Add document APiEcMLBQKmPgHr5kenv > xhbpBk11568rLJqKv5gS	+ Start collection + Add field TipsnTrick_ID: 2 TipsnTrick_Title: "Pet in Condos: 5 Key considerations" TipsnTrick_Content: "Pet owners look for a condo to live in, they look for pet-friendly buildings. So when a building prohibits pets you reduce your prospective owners and tenants who want to live there. Your condo may also face criticism as it may be considered discriminatory to prohibit pets. Just like people, dogs have their own personalities that influence their behavior. Some dogs may be more outgoing and confident, while others may be more timid and anxious. Different dog breeds can have certain behaviors due to selective breeding. For example, herding breeds, such as sheepdogs, may like to herd other animals. However, this does not mean all dogs of a breed will show the same behavior, as"

Figure 15: Cloud Firebase - Tips and Tricks

The screenshot shows the Cloud Firestore interface. The left sidebar lists collections: (default), Pet appointment, Admin, Pet, Pet appointment (selected), Tips and Trick, User, and images. The main area shows the Pet appointment collection with one document selected, identified by its ID: HLUjhiaQSPFN1q8UHRIS. This document contains the following fields and values:

- Appt_Date: "2024-05-13"
- Appt_ID: "29ca3c83-33c0-4dfd-82f0-057fc5329ee0"
- Appt_Location: "Rak Na Chan Clinic"
- Appt_Note: "x-ray and heart echo"
- Appt_Pet: "Chanel"
- Appt_Time: "12:00 PM"
- Appt_Type: "Veterinary"
- Pet_ID: "oiwefjuwahiaerghuahao"
- User_ID: 6

Figure 16: Cloud Firebase - Pet appointment

- Programming and Scripting Tools
 - o Dart Programming language

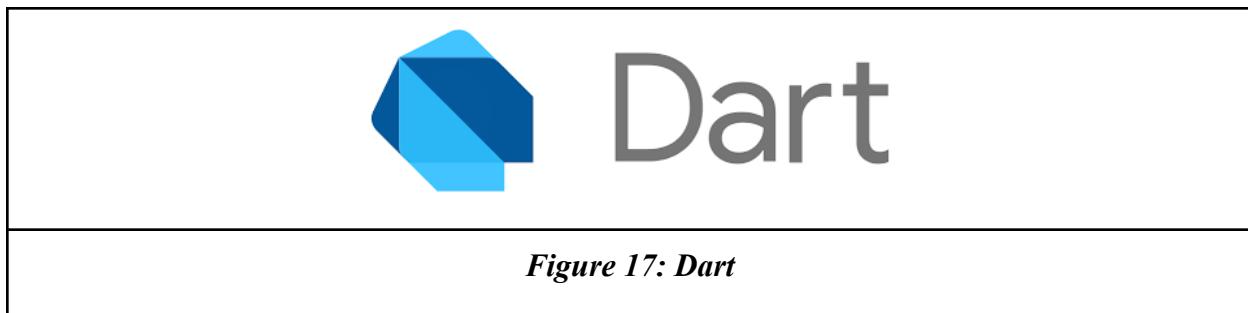


Figure 17: Dart

The primary programming language to develop this project is Dart. Dart is a client development programming language developed by Google, primarily for building web, mobile, and desktop applications [8]. Dart libraries such as `dart:core`, `Stream` (`dart:async`), and (`dart:io`), support several programming features of this project. For instance, `Stream` (`dart:async`) is used for working with sequences of asynchronous events which enables handling asynchronous data sources, such as user input, network responses, or timer events, in a declarative and efficient manner.

- o Visual Studio

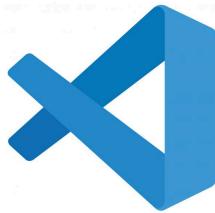


Figure 18: Visual Studio Code

The source code editor chosen for this project is Visual Studio Code (VS Code). We opted for VS Code due to its versatility and compatibility with both Windows and macOS [9], which are the operating systems used by our group members. The main reason for selecting VS Code is its effective support for programming languages, including Dart and Flutter, which are essential for our Flutter project. Additionally, VS Code offers powerful collaboration features such as Live Share, allowing team members to work together in real-time, which has been crucial for our group project [9].

Components

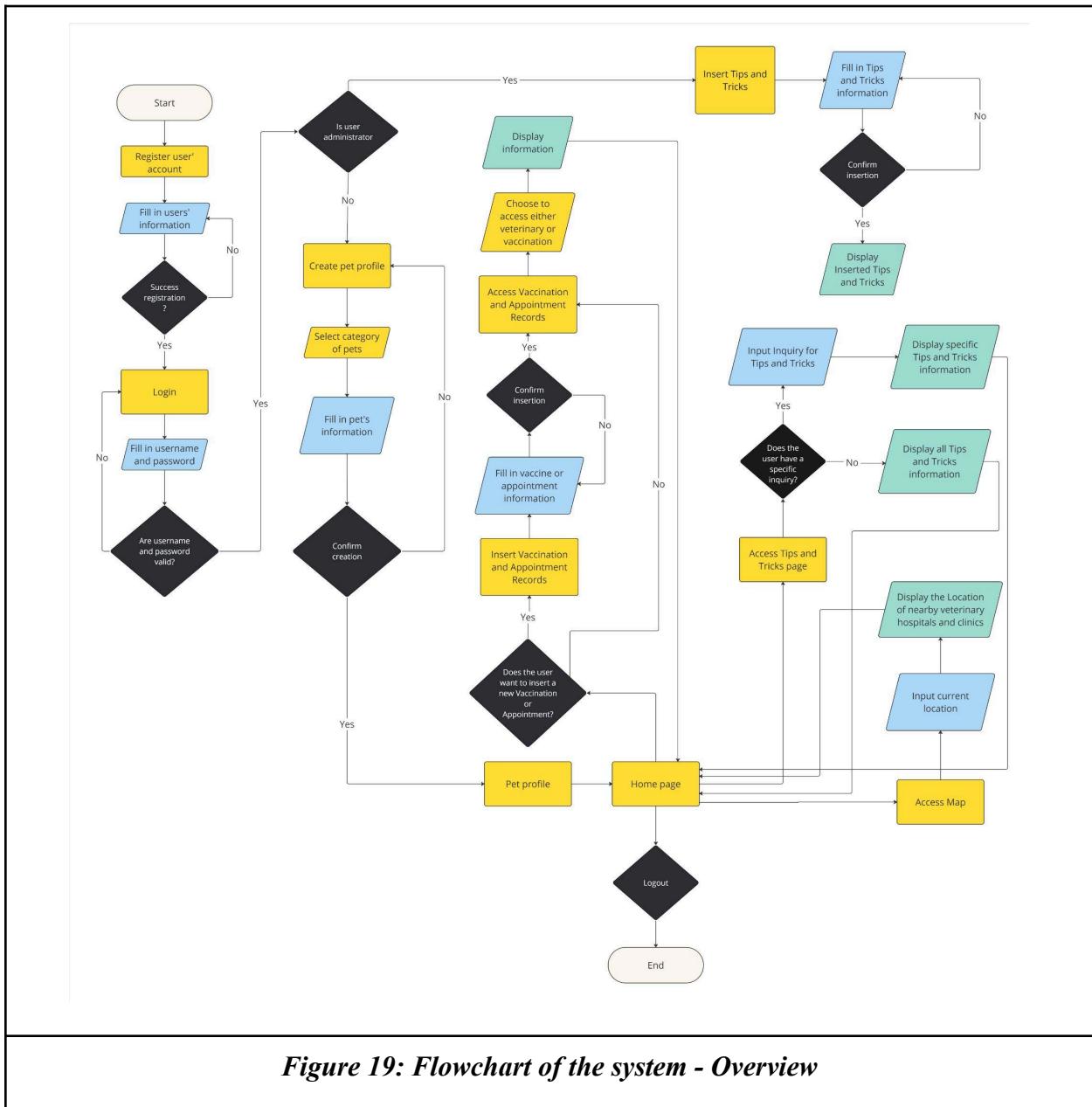


Figure 19: Flowchart of the system - Overview

According to Figure 19, this section illustrates the component of our application “Happy Tails” as a Flowchart to explain the overall system components and workflow. The process commences with users establishing their own account through registration, where users provide the necessary information and login with a valid username and password. After successfully logging in, the system verifies the user_id against the administrator database to ensure it does not match and proceeds to initiate the pet profile creation process. Users can navigate to the Home page which is the primary user interface providing access to various other pages including

Vaccination and Appointment Records, Tips and Tricks page, and Access Map. The Home page serves as a central hub, facilitating exploration and different features within the application. Administrators possess the ability to input and manage information in the Tips and Tricks section. When users decide to conclude their session on Happy Tails, they can log out from their account using the user profile page.

4.2 Implementation Guide and Techniques

4.2.1 Connecting Guide

In order to implement this project, the application requires additional components or plugins including FlutterFire CLI to configure the Flutter application to connect to Firebase. Initially, to initialize Firebase in the Flutter application, it is necessary to run ‘flutter pub add firebase_core’ which is a Flutter plugin that utilizes the Firebase Core API and enables connecting to multiple Firebase apps. Afterward, ‘flutterfire configure’ command is required to be run in the app configuration workflow of the project, helping the Firebase app to run on different platforms and ensuring that the Firebase configuration is up-to-date.

4.2.2 Techniques

This project utilizes the technology of Google Maps API which displays the map within the application. To begin with, it is required to obtain an API key at <https://cloud.google.com/maps-platform/> so that the application will be able to build the Google Maps widget. Subsequently, to update the dependency in pubspec.yaml, it is required to install the Google Maps plugin by running ‘flutter pub add google_maps_flutter’ and install the location plugin by running the command ‘flutter pub add location’. As a result, installing these commands enables Google Map SDK for each platform and location handles fetching a location on Android and iOS which is beneficial in getting the current location of the users.

```

import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart'; //flutter pub add google_maps_flutter
import 'package:location/location.dart'; //flutter pub add location

Future<bool> requestLocationPermission() async {
  Location location = Location();
  PermissionStatus permissionStatus = await location.hasPermission();
  if (permissionStatus == PermissionStatus.denied) {
    permissionStatus = await location.requestPermission();
  }
  return permissionStatus == PermissionStatus.granted;
}

Future<LocationData?> getCurrentLocation() async {
  Location location = Location();
  bool hasPermission = await requestLocationPermission();
  if (hasPermission) {
    return await location.getLocation();
  } else {
    return null;
  }
}

```

```

@Override
Widget build(BuildContext context) {
  return Scaffold(
    body: currentLocation == null
      ? Center(child: CircularProgressIndicator())
      : GoogleMap(
        onMapCreated: (controller) {
          mapController = controller;
        },
        initialCameraPosition: CameraPosition(
          target: LatLng(
            currentLocation!.latitude!,
            currentLocation!.longitude!
          ), // LatLng
          // target: _center,
          zoom: 14,
        ), // CameraPosition
        markers: Set<Marker>.of(_markers.values),
      ), // GoogleMap
)

```



The implementation

The output

Figure 20: Google Maps API

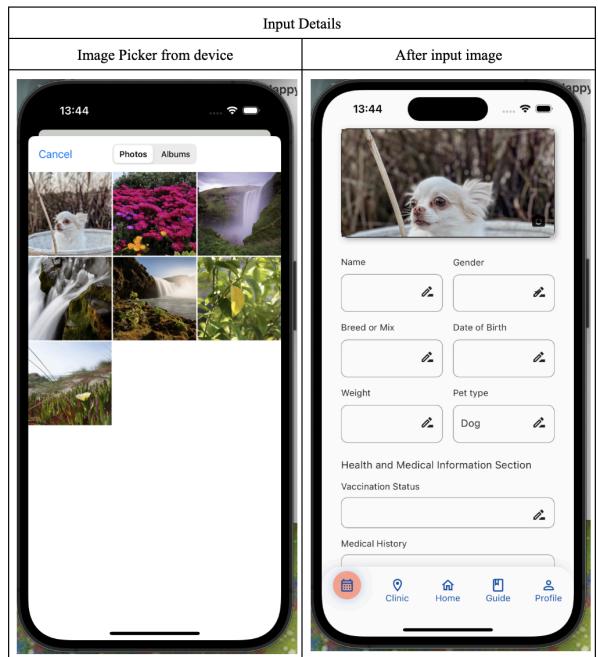
Moreover, the project incorporates the 'Image Picker plugin' for selecting images from the image library based on the instructions from https://pub.dev/packages/image_picker. In order to activate this library, first install it using the 'flutter pub add image_picker' command. This command will add 'image_picker: ^1.0.7' to the dependencies of the Flutter project. Next, import 'package:image_picker/image_picker.dart' in the Dart code of the file where intend to enable the image picker functionality. Consequently, this technique will enable them to input the image from the gallery into the application. For example, as shown in Figure 20, this project implemented this technique in creating a pet profile to enable users to input their pet image.

```

void selectedImage() async {
  Uint8List img = await pickImage(ImageSource.gallery);
  if (img != null) {
    setState(() {
      _img = img;
    });
  }
}

child: Stack(
  alignment: Alignment.bottomRight,
  children: [
    if (_img != null)
      Image.memory(
        _img!,
        width: double.infinity,
        height: double.infinity,
        fit: BoxFit.cover,
      ), // Image.memory
    IconButton(
      icon: Icon(Icons.camera_alt),
      // onPressed: _pickAndNavigateToNextPage,
      onPressed: () {
        selectedImage();
      },
    ), // IconButton
  ],
), // Stack
), // Container

```



The implementation

The output

Figure 21: image_picker

CHAPTER 5

TESTING AND EVALUATION

5.1 Unit Tests

For the unit tests, the team selected some important and critical processes for formal unit testing. The selected processes include:

- Process No. 1: Register
- Process No. 2: Login
- Process No. 3: Create a pet profile
- Process No. 4: Insert Vaccination and Appointment Records
- Process No. 5: Insert Vaccination and Appointment Records

5.1.1 Test Performed on Process No. 1: Register

The register process contains 4 input fields including Full name, Username, Password, and Confirm Password.

Table 5.1: Test on Process Register

Operation Performed	Condition Tested	Actual Result
A warning message: All fields – Users submit nothing to the system	The system displays a warning message under every empty box to inform users to enter their information accordingly.	Pass
A warning message: Username field – Users submit a username that already exists in the system.	The system shows a warning message box below the screen to notify users that the username already exists in the system: “User already exists”	Pass
A warning message: Password field – Users submit a password that is less than 6 characters.	The system displays a warning message under the Password field to inform users that the password’s length should not be less than 6 characters “Password must be at least 6 characters”	Pass
A warning message:	The system displays a warning message under the Confirm Password field to	Pass

Confirm Password field – Users submit a confirmed password that does not match the Password field.	inform users that the user must confirm the password that matches the password in the previous field “Passwords do not match.”	
A warning message and Page routing: Users submit all appropriate registered information to the system	The system displays a message informing the user that the registration is completed “Register successfully” and redirects the user to the Login page	Pass

5.1.2 Test Performed on Process No. 2: Login

The login process contains 2 input fields including Username, Password, and 1 radio button to Agree to terms of use.

Table 5.2: Test on Process Login

Operation Performed	Condition Tested	Actual Result
A warning message: All fields – Users submit nothing to the system	The system displays a warning message under every empty box to inform the user to enter their username and password accordingly.	Pass
A warning message: Username field – Users submit a username that does not match any record in the system.	The system shows a warning message box below the screen to notify users that the username is invalid: “Invalid email or password”	Pass
A warning message: Password field – Users submit a password that does not match the given username.	The system shows a warning message box below the screen to notify users that the username is invalid: “Invalid email or password”	Pass

A warning message: Agree to terms of use field – Users do not click on the radio button to agree to the terms of use.	The system shows a warning message box below the screen to notify users to check the radio button to agree to the terms of use: “Please agree to the terms”	
Page routing: Users submitted all appropriate login information to the system.	The system redirects the user to the Welcome page.	Pass

5.1.3 Test Performed on Process No. 3: Create a pet profile

The create pet profile process contains 11 input fields including the Pet’s image, Name, Gender, Breed or mix, Date of Birth, Weight, Vaccination status (optional), Medical history (including any chronic conditions) (optional), Allergies (optional), Medication (if applicable), and Doctor appointment (optional).

Table 5.3: Test on Process Create a Pet Profile

Operation Performed	Condition Tested	Actual Result
Page routing: Pet-type selection page – Users select a type of pet that they want to create a profile for and click the Confirm button.	The system redirects the user to the Create Pet Profile page.	Pass
A warning message: Name, Gender, Breed or mix, Date of Birth, and Weight fields – Users submitted nothing to the system	The system displays a warning message under every empty box to inform the user to enter their pet’s information accordingly.	Pass
A warning message: Pet’s image fields – Users did not select the pet’s image.	The system shows a warning message box below the screen to notify users by “No selected image” and the system will set the pet image as a default image.	Pass

Page routing: Pet type field – Users want to edit their pet type by clicking the pen icon.	The system redirects to the pet-type selection page.	Pass
Page routing: Users submitted all appropriate pet profile information to the system	The system redirects the user to the Pet Profile page.	Pass

5.1.4 Test Performed on Process No. 4: Insert Vaccination and Appointment Records

The Insert Vaccination and Appointment Records process contains 6 input fields including the Date of Appointment, Time of Appointment, Type of Appointment, Pet, Location, and Note (optional).

Table 5.4: Test on Process Insert Vaccination and Appointment Records

Operation Performed	Condition Tested	Actual Result
A warning message: Date of Appointment, Time of Appointment, Type of Appointment, Pet, Location fields – Users submitted nothing to the system	The system displays a warning message under every empty box to inform the user to enter appointment information accordingly.	Pass
Page routing: Users submitted all appropriate Insert Vaccination and Appointment information to the system	The system redirects the user to the All Appointments page.	Pass

5.2 System Integration Test

This topic aims to visualize the actual result of the implementation to ensure its functionalities according to the required functions.

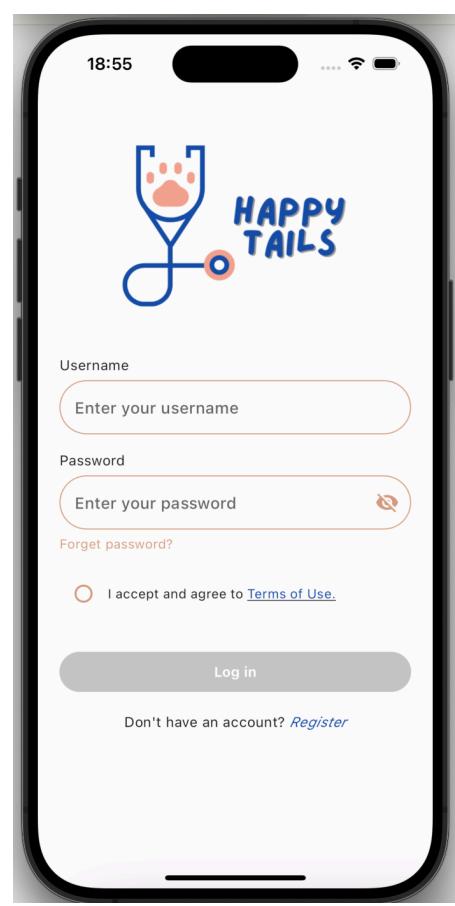
5.2.1 Test Scenario

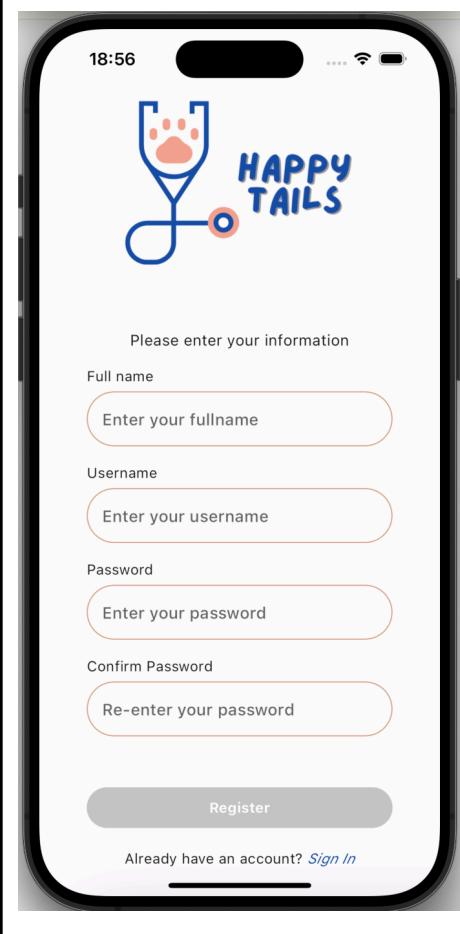
To test all functional aspects of the system thoroughly, the researcher set up a test scenario that consisted of 6 phases as shown below.

- Registration: Users register to the system.
- Login: Users log in to the system.
- Create a pet profile: Users create a pet profile to record their pet's information.
- Insert Vaccination and Appointment Records: Users create a new Vaccination and Veterinary (Appointment) record.
- Access Tips and Tricks: Users access the Tips and Tricks page.
- Access Map: Users access the Nearby Clinics page.

In addition, the test scenario can be used as a user guideline as it covers all the steps necessary to use the system. The details of each phase are displayed in the next section.

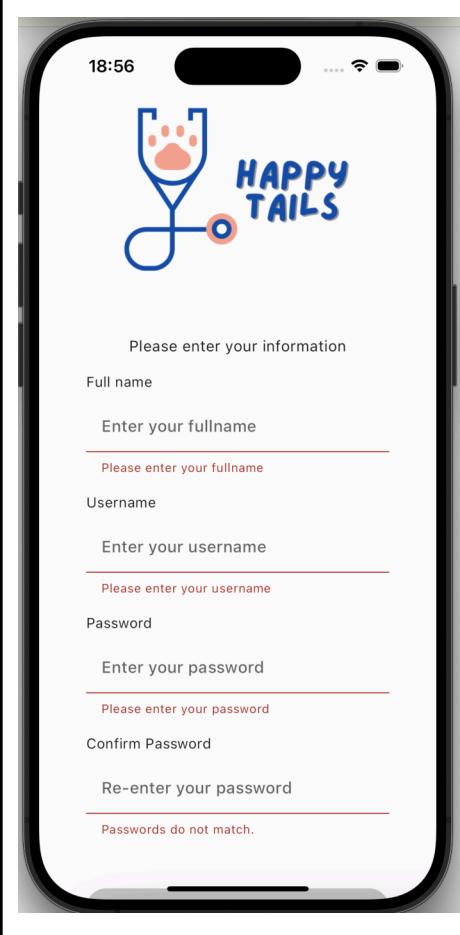
5.2.1.1 Registration: Users register to the system

	<p>The phase starts with the user landing on the loading page of the application and then the login page.</p> <ul style="list-style-type: none">• In this scenario, the user who enters the application for the first time must register an account before logging into the system.• The user clicks on the text Register below the “Log in” button.
--	--

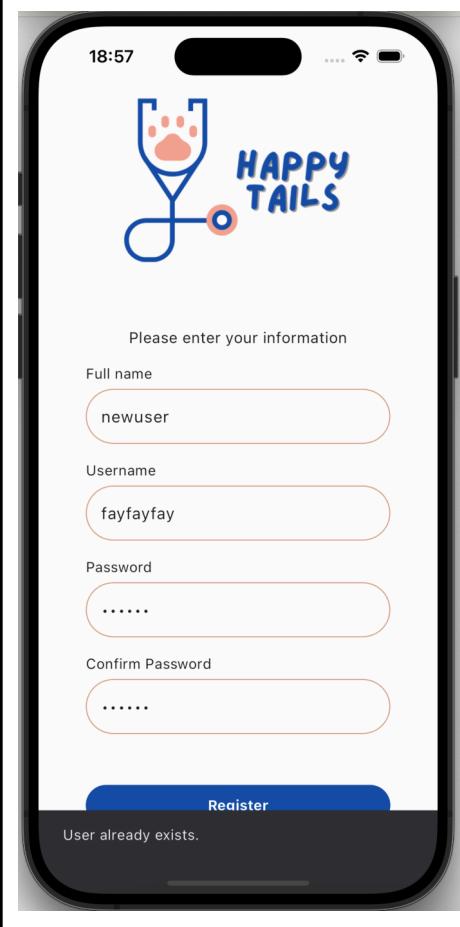


After the user clicks Register, it will navigate the user to the register page.

- The Register page has 4 input fields including Full name, Username, Password, and Confirm Password.
- Below the 4 input fields, it has a “Register” button and the text allowing the user to return to the login page.
- After filling in all input fields, users can click the “Register” button to register to the system.

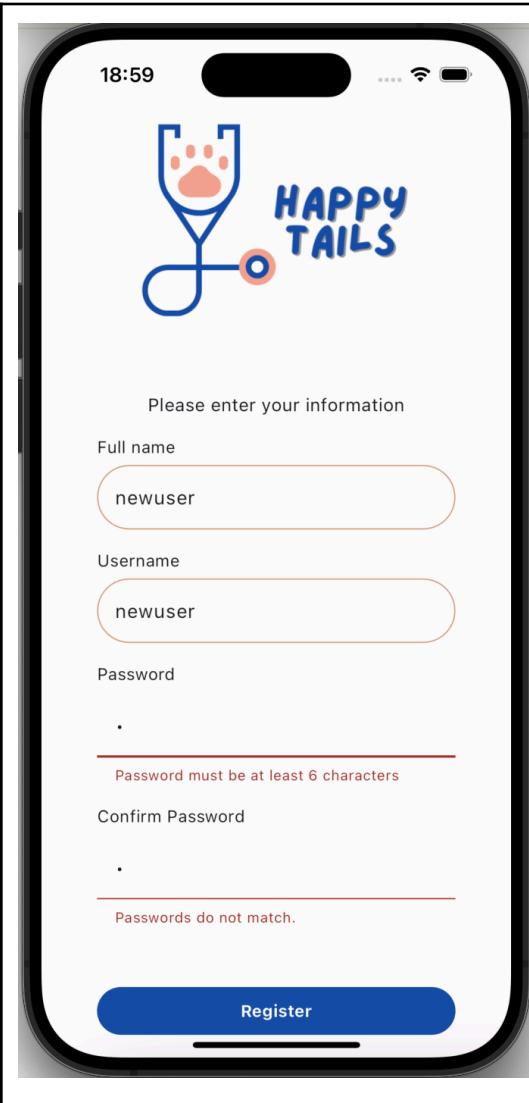


If the user clicks "Register" without entering any information, a warning notice will display beneath each input field.



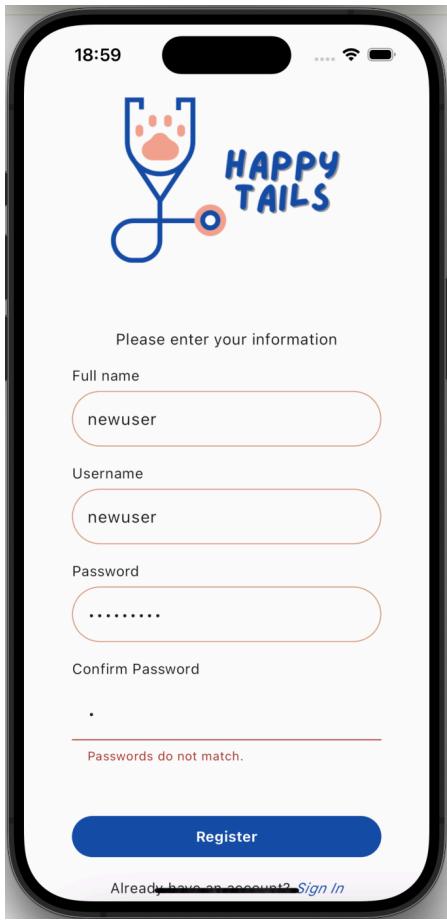
If the user submits a username that already exists in the system, a warning notice will display below the screen.

- In this scenario, the user enters username = “fayfayfay” which already exists in the system.



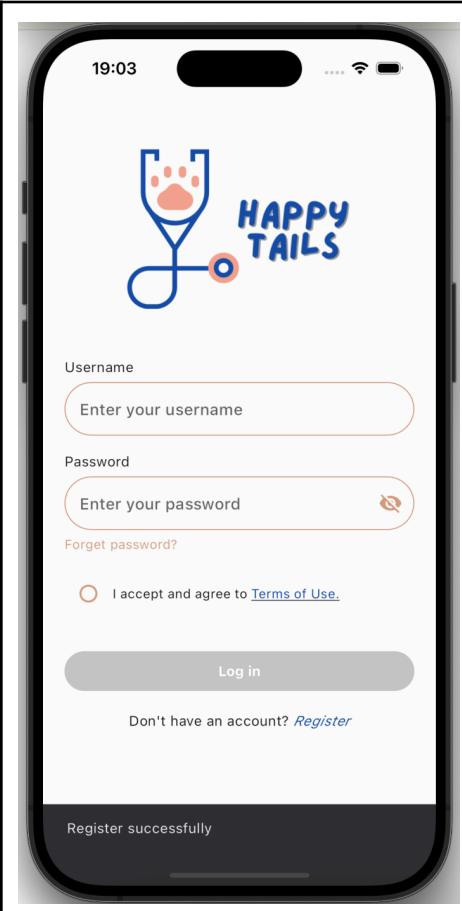
If the user submits a password that is less than 6 characters, a warning notice will display under the password field.

- In this scenario, the user enters password = “1” which is shorter than 6 characters.



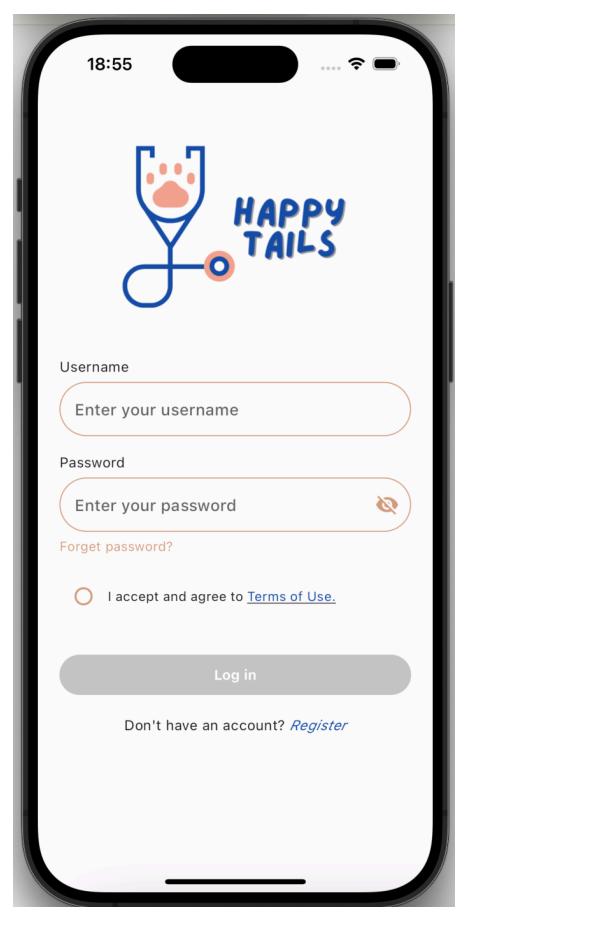
If the user submits a confirmed password that does not match the password in the Password field, a warning notice will display under the Confirm password field.

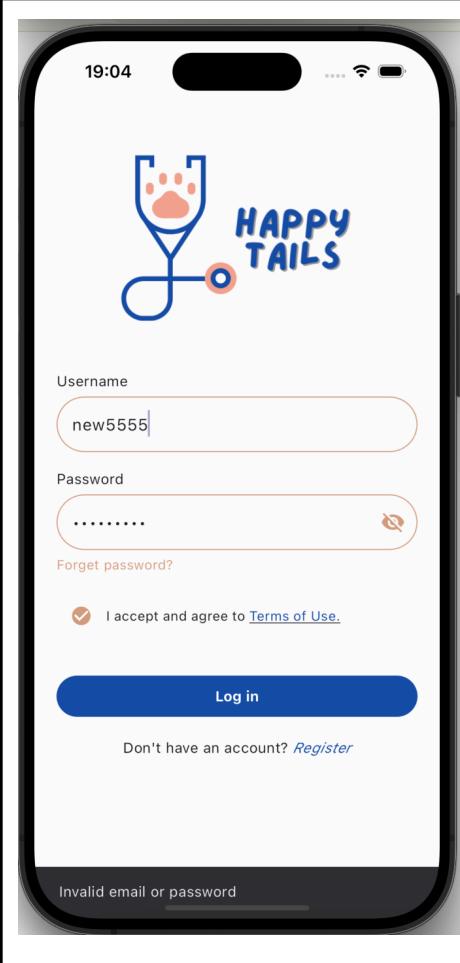
- In this scenario, the user enters password = “newnewnew” and the confirm password = “n” which is mismatched.



If the user submits all appropriate registered information to the system, a notice will appear at the end of the screen “Register successfully” and the system will redirect the user to the Login page.

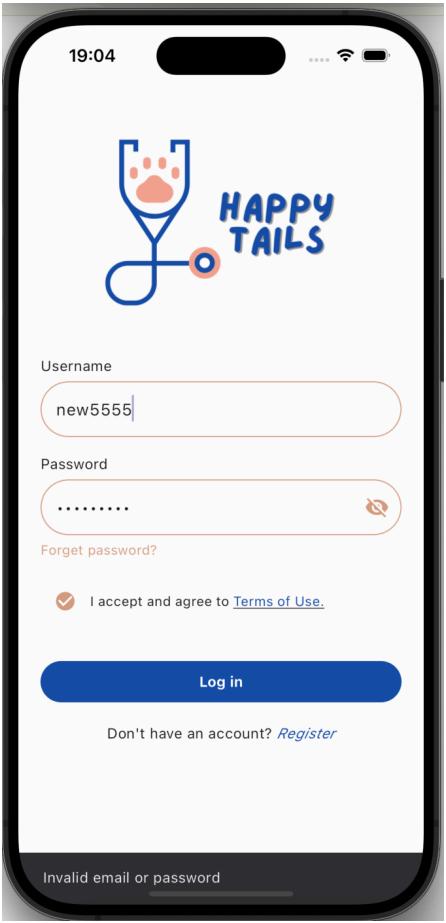
5.2.1.2 Login: Users log in to the system

	<p>The phase starts with the user landing on the login page.</p> <ul style="list-style-type: none">• If the users have already registered an account in the register phase, they can log in via this login page.• The Login page has 2 input fields including Username, Password, and 1 radio button to Agree to terms of use.• Below the input fields, it has a “Login” button and the text allowing the user to return to the register page.• After filling in all input fields, users can click the “Login” button to log in to the system.
--	---



If the user submits a username that does not match any record in the system, a warning notice will display below the screen.

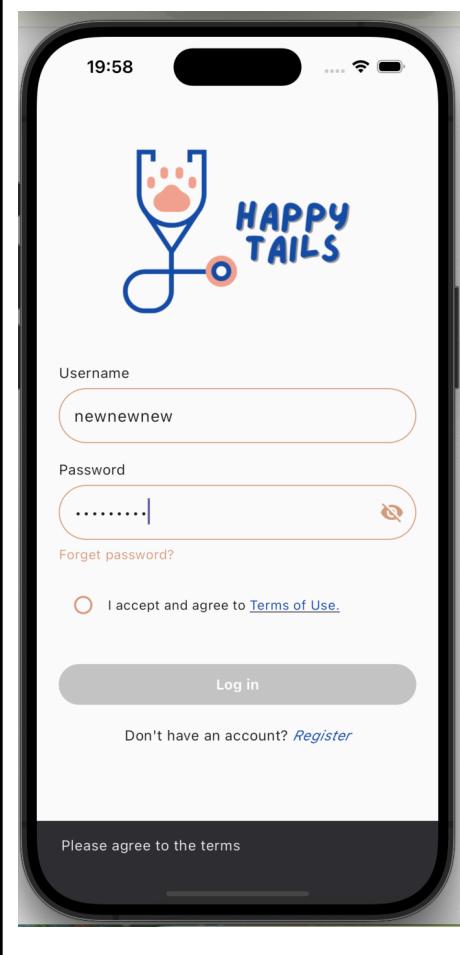
- In this scenario, the user enters username = “new5555” which does not exist in the system.



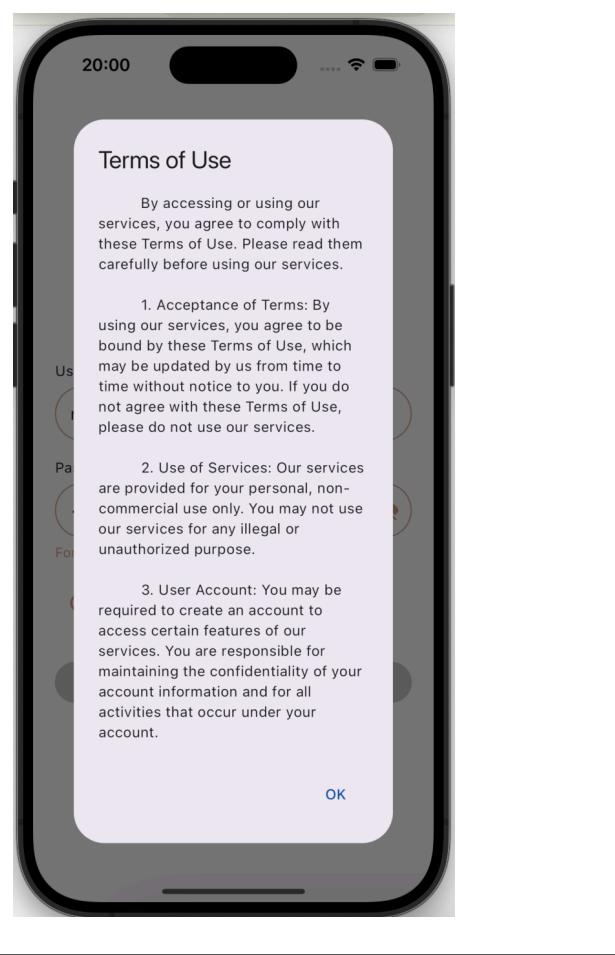
The screenshot shows the login screen of a mobile application. At the top, there is a logo featuring a blue stethoscope with a red paw print on its earpiece, next to the text "HAPPY TAILS". Below the logo are two input fields: "Username" containing "new5555" and "Password" containing ".....". There is a "Forgot password?" link and a checkbox for accepting terms. A large blue "Log in" button is at the bottom, with a "Register" link below it. At the very bottom of the screen, a faint message "Invalid email or password" is visible.

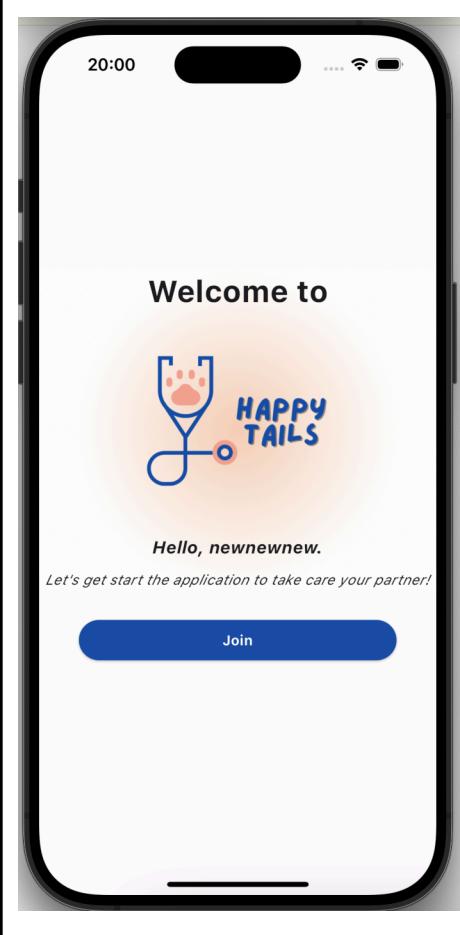
If the user submits a password that does not match the given username in the Username field, a warning notice will display below the screen.

- In this scenario, the user enters username = “new5555” and password = “newnew555” which do not match.



If the user does not click on the radio button to agree to the terms of use, a warning notice will display below the screen.

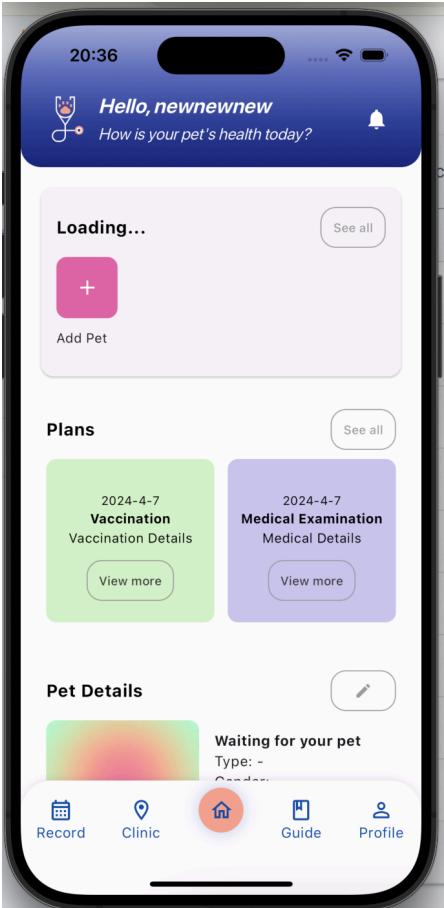
	<p>If the user clicks on Terms of Use, the system will present a Terms of Use of the application.</p> <ul style="list-style-type: none"> Once the user completes reading the terms of use, the user can click the “OK” button to close this popup.
--	---



If the user submits all appropriate login information to the system, the system will redirect the user to the Welcome page.

- The Welcome page displays the username the user used to log in and the user can click on the “Join” button to enter the application.

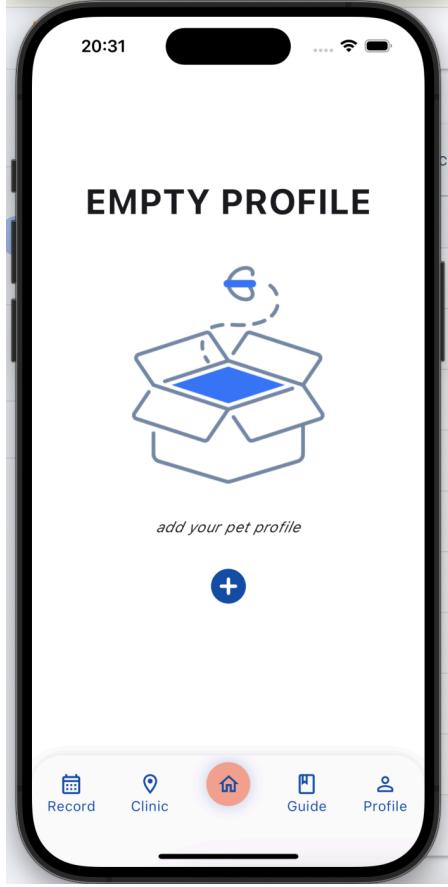
5.2.1.3 Create a pet profile: Users create a pet profile to record their pet's information



The screenshot shows a mobile application interface. At the top, there is a blue header bar with the text "Hello, newnewnew" and "How is your pet's health today?". Below the header, the screen displays a "Loading..." message with a pink plus button labeled "Add Pet". To the right of this is a "See all" button. The next section, "Plans", contains two items: a green box for "2024-4-7 Vaccination" with "Vaccination Details" and a purple box for "2024-4-7 Medical Examination" with "Medical Details", both with "View more" buttons. Below these is a "Pet Details" section showing a colorful gradient bar, the text "Waiting for your pet", and "Type: -". At the bottom of the screen are five navigation icons: "Record" (calendar), "Clinic" (location pin), a central orange "Home" icon, "Guide" (book), and "Profile" (person).

The phase starts with the user landing on a Homepage.

- If the user who logged in has not created any pet profile in the application, the first section that shows a list of pets will be empty with a button to add a pet.
- If the user logged in has created a pet profile in the application, the first section that shows a list of pets will show a list of pets with a button to add a pet.

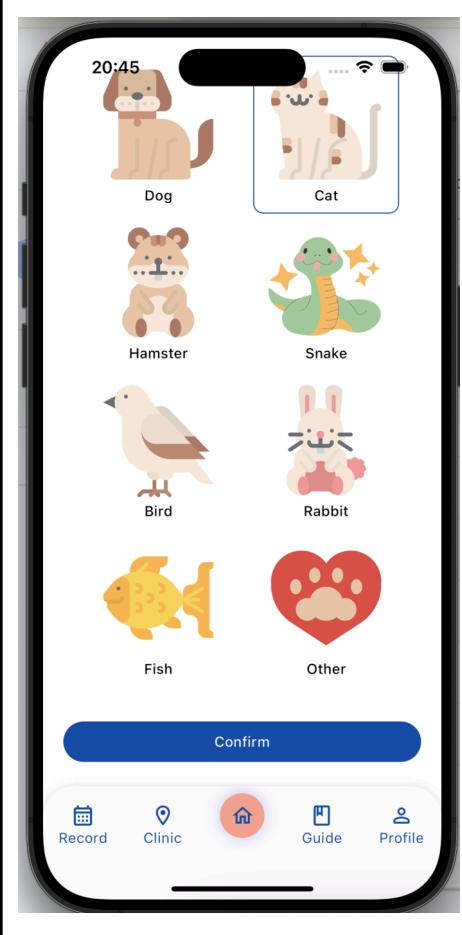


The image shows a smartphone screen displaying an application interface. The title "EMPTY PROFILE" is at the top. Below it is a large blue icon of an open box with a small dog inside. The text "add your pet profile" is written below the icon. A blue circular button with a white plus sign is centered below the text. At the bottom of the screen is a navigation bar with five items: "Record" (calendar icon), "Clinic" (location pin icon), "Guide" (house icon, highlighted in orange), and "Profile" (person icon).

- In this scenario, the user wants to create a pet profile, and the system will bring the user to a create pet profile page showing an add button.

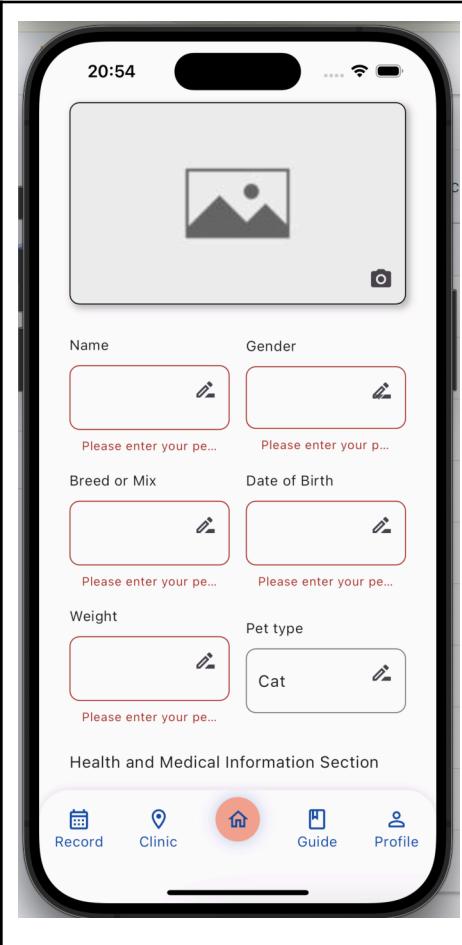


- After clicking the Add button, the user will be redirected to the option pet page where the user can select the pet type from this page.

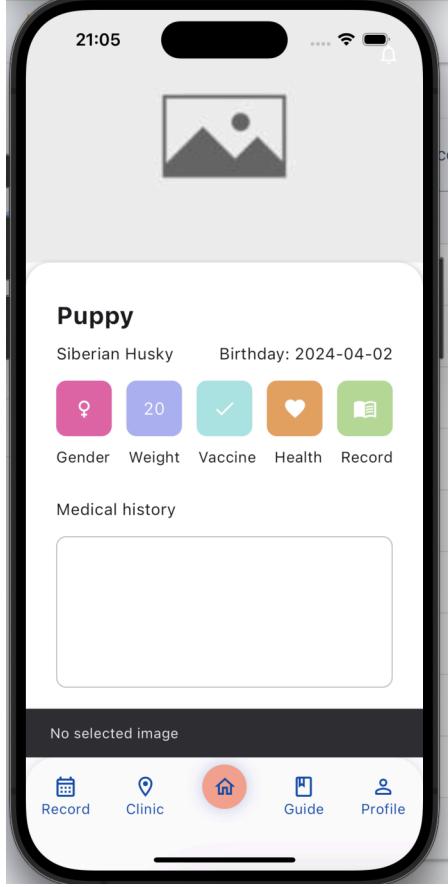


- Once the user has selected a type of pet he or she wants to create a profile for, the user can click on the “Confirm” button to navigate to the next page.

The Create Pet Profile page contains 11 input fields including the Pet's image, Name, Gender, Breed or mix, Date of Birth, Weight, Vaccination status (optional), Medical history (including any chronic conditions) (optional), Allergies (optional), Medication (if applicable), and Doctor appointment (optional).

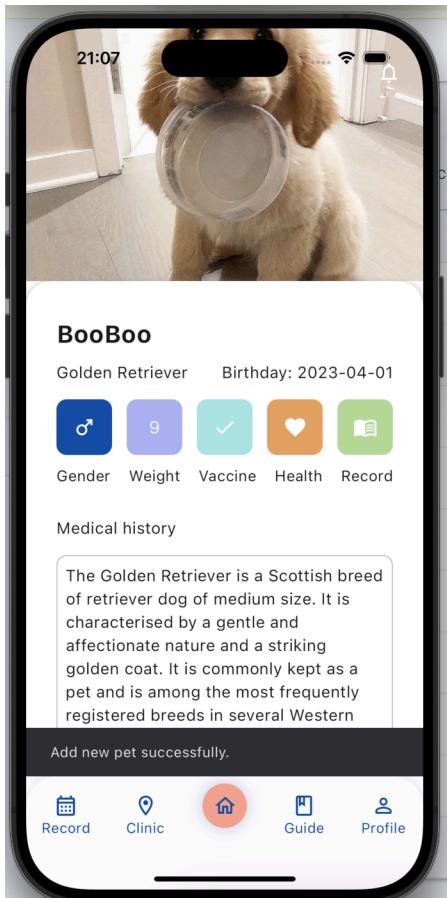


- If the user does not input name, gender, breed or mix, date of birth, or weight, a warning notice will display under the field that does not have input.

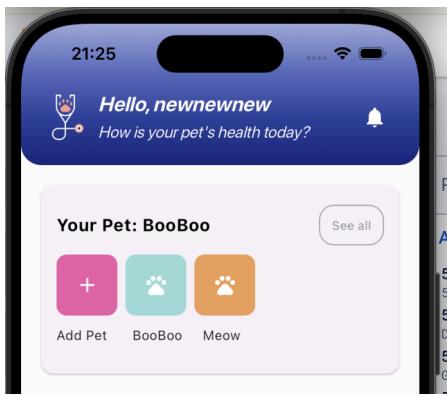


The screenshot shows a mobile application interface for a pet named 'Puppy'. At the top, there is a placeholder for a pet image with a small sun icon. Below it, the pet's name 'Puppy' is displayed in bold, followed by its breed 'Siberian Husky' and birthday '2024-04-02'. There are five colored buttons below this information: pink (Gender), light blue (Weight), teal (Vaccine), orange (Health), and green (Record). A section titled 'Medical history' follows, containing a large empty rectangular box. At the bottom of the screen, a dark bar displays the message 'No selected image'. The navigation bar at the very bottom includes icons for 'Record' (calendar), 'Clinic' (location pin), 'Guide' (orange circle with house icon, highlighted in red), and 'Profile' (person).

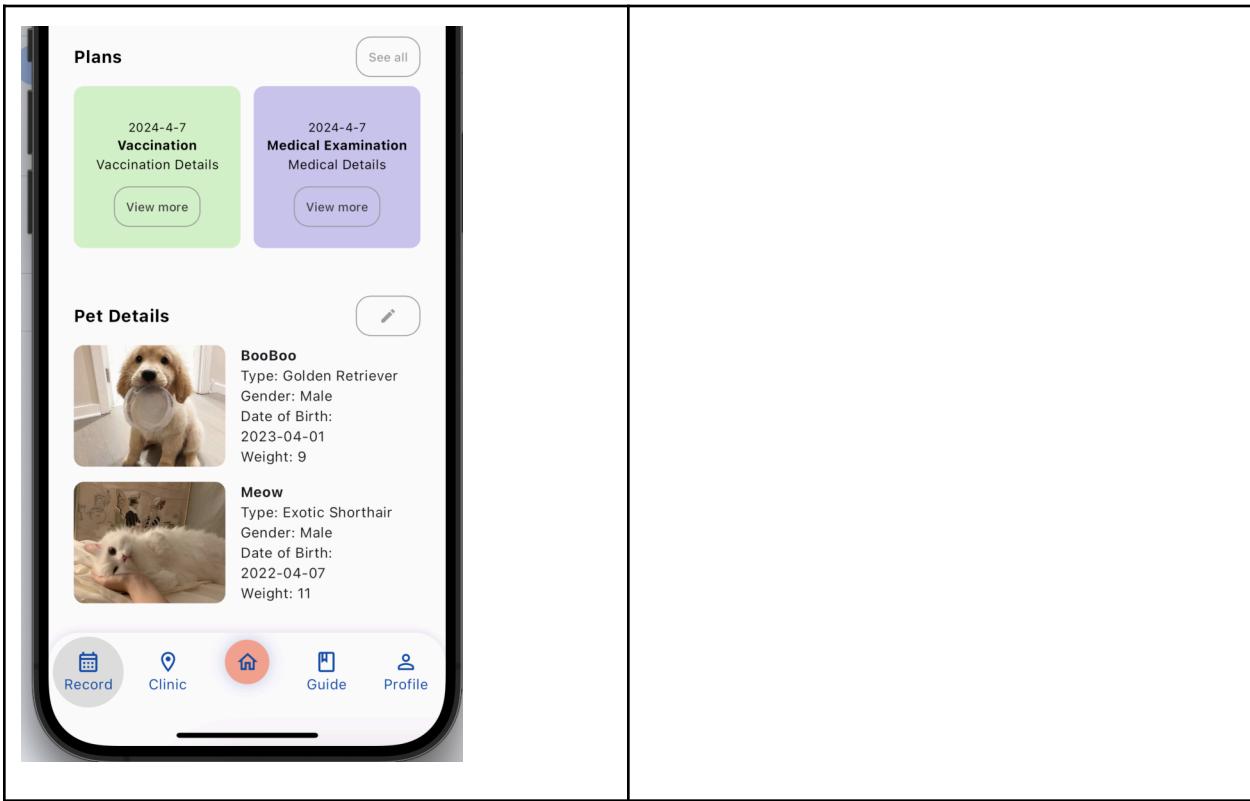
If the user clicks the ‘Confirm’ button without selecting a pet image, a warning notice will display below the screen.



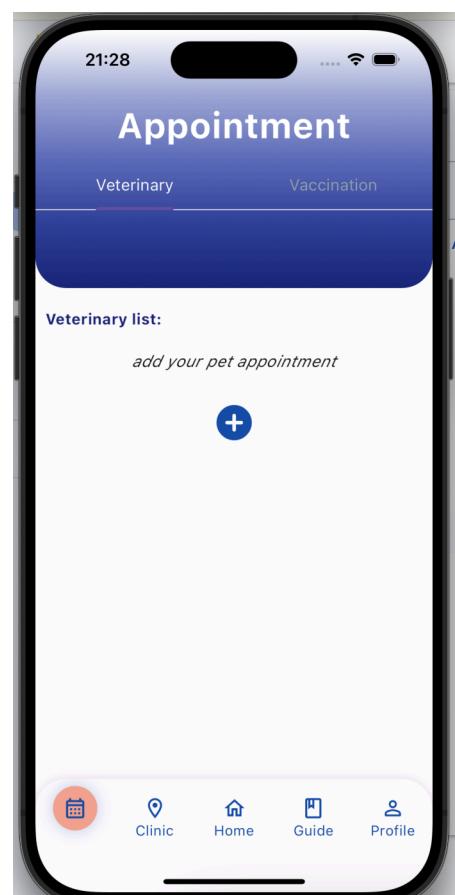
If the user completely fills out the pet information and clicks ‘Confirm’, the system redirects the user to the pet profile page and shows a notice below the screen indicating “Add new pet successfully.”

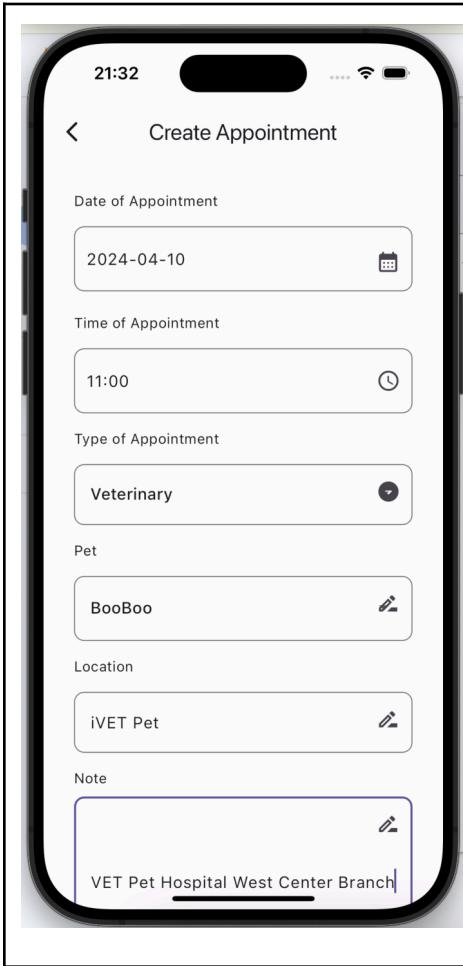


After successfully creating a pet and then going back to the homepage, the user will see a pet that the user added on the homepage.



5.2.1.4 Insert Vaccination and Appointment Records: Users create a new Vaccination and Veterinary (Appointment) record.

	<p>The phase starts with the user landing on the Appointments page of the application.</p> <ul style="list-style-type: none">• The user can click on the App bar at the top to select the type of appointment to view (veterinary or vaccination).• If the user does not have an appointment history with a type of veterinary, the veterinary appointment page will display an empty list of appointments with the Add button navigating to the Create Appointment page.• If the user does not have an appointment history with a type of vaccination, the vaccination appointment page will display an empty list of appointments with the Add button navigating to the Create Appointment page.
--	--



After the user clicks the add button, the system navigates the user to the Create Appointment page which contains 6 input fields including the Date of Appointment, Time of Appointment, Type of Appointment, Pet, Location, and Note (optional).

- The user is required to enter the date, time, type, pet name, and location details before clicking the “Confirm” button to add the appointment.

21:29

< Create Appointment

Date of Appointment

Please select appointment date

Time of Appointment

Please select appointment time

Type of Appointment

Please select the appointment type

Pet

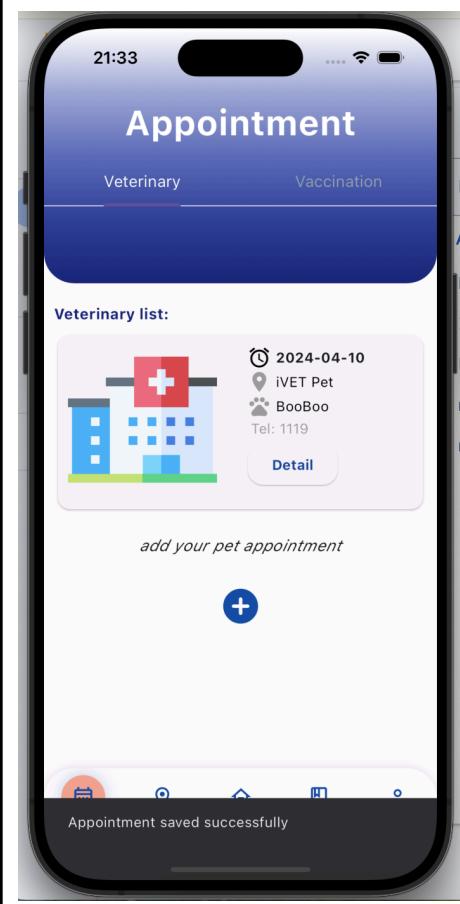
Please select your pet

Location

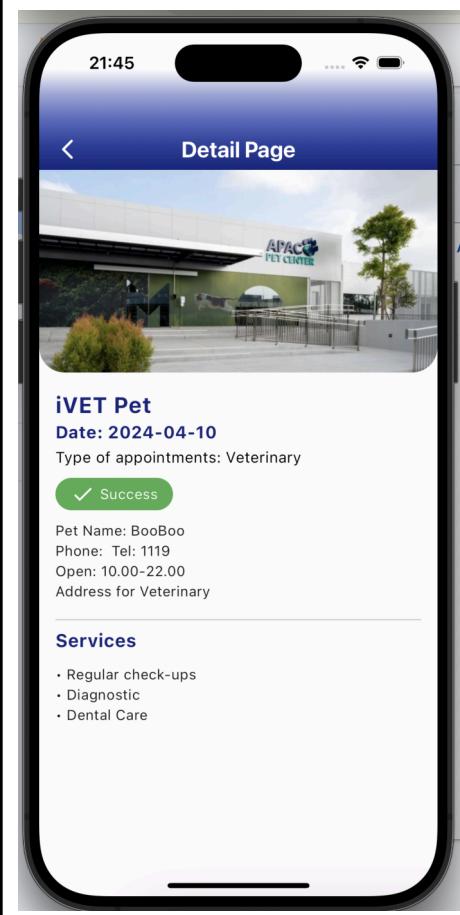
Please enter appointment location

Note

If the user does not enter any information, the system displays a warning message under every input field except the Note field to inform the user to enter the appointment information.



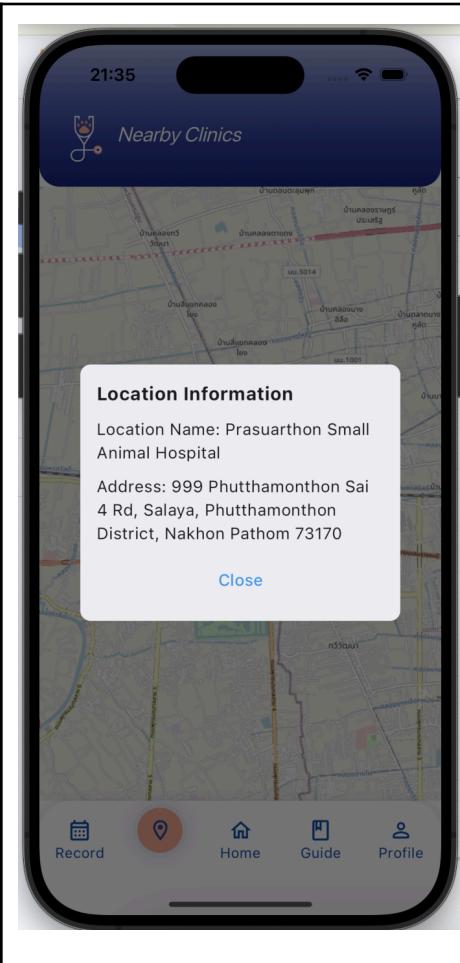
If the user submits all appropriate Appointment information to the system, the result of the created pet appointment will be displayed on the Appointment List page based on the selected type.



If the user clicks 'Detail' from the pet appointment, the system redirects the user to the detail appointment page where the user can see the details of the appointment.

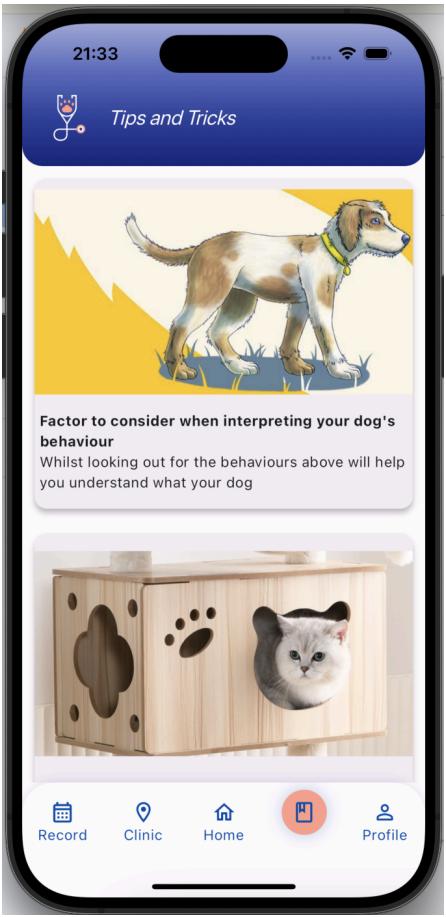
5.2.1.5 Access Map: Users access the Access Map

	<p>The phase starts with the user landing on the Nearby Clinic page of the application.</p> <ul style="list-style-type: none">• The user can access the Nearby Clinic page by clicking the “Clinic” button on the navigate bar.
--	---



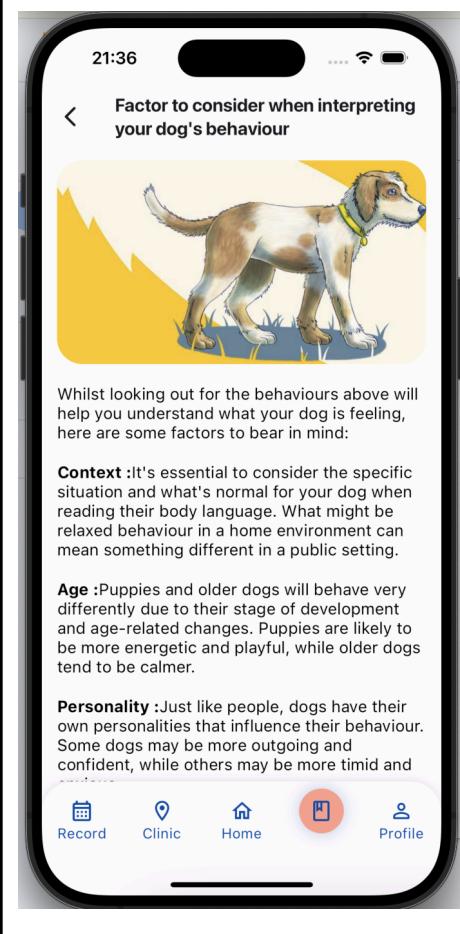
- The user can view the map of his or her nearby clinic location being pinned on the map.
- The user can click on each pin to view the information about the clinic including clinic name and clinic location.

5.2.1.6 Access Tips and Tricks: Users access the Tips and Tricks page



The phase starts with the user landing on the Tips and Tricks page of the application.

- The user can access the Tips and Tricks page by clicking the “Guild” button on the navigate bar.



- After clicking on the card of trips and tricks, the user can access complete information.
- The user can return to the full Tips and Tricks page by clicking the return button on the top left of the page.

CHAPTER 6

CONCLUSIONS

6.1 Benefits

6.1.1 Benefits to Project Developers

- Developing Happy Tails provides developers with an opportunity to improve their programming skills and expand their knowledge of database management, UI design, and mobile app development, especially Dart and Flutter programming.
- It allows developers to obtain additional knowledge of implementing external resources with the project such as Google Map API, OpenStreetMap, Firebase, and many more.
- Building the Happy Tails application serves as a valuable addition to developers' portfolios, showcasing their proficiency in developing real-world applications along with their knowledge of programming and UX/UI design.
- Implementing this project in a team fosters collaboration and teamwork skills among developers, helping them achieve and improve their communication skills along with other technical skills.

6.1.2 Benefits to Users

- The Happy Tails application allows users to easily manage their pets' information and appointments, allowing them to stay organized and ensure the well-being of their pets.
- The Happy Tails application is convenient and accessible to individuals of any age, providing pet owners with a more convenient method to monitor their pets' health by centralizing pet-related information and services in one app.
- The Happy Tails application provides helpful tips and information to aid in users' understanding of basic pet care and the well-being of their pets. This will decrease the likelihood of mistakes in animal care and enhance new pet owners' comprehension of how to take care of their pets.
- The users can access the map to obtain information regarding nearby clinics.

6.2 Problems and Limitations

- Importation of images into the Firebase database and retrieving data in Firebase through flutter code can be bothersome since Cloud Firestore does not have a data type to record an image.
 - Thus, the developers have resorted to using base64 as a method to encode the image uploaded by users; however, testing the application on the simulator does not allow encoding of base64 forcing the developers to shift to the method of storing the uploaded image on Cloud storage of Firebase and store those image's directory path on the Cloud Firestore.
- Limitation on Google Map API free quotas: Google Maps offers a limitation on free usage, with additional usage subject to a monthly fee.
 - Therefore, the developers have resorted to the method of switching the platform to obtain a map interface from Google Maps to OpenStreetMap which is free to all users with an interface that is less detailed than Google Maps.
- Testing on the real phone: the testing process on real phones or simulators can be time-consuming due to manual installation, freezing or performance issues, and prediction of interface based on different screen sizes.

6.3 Future Work

- The notification system: this function will notify the user when approaching the pet appointment schedules.
- The admin interface: the interface allows staff to access and manage users' information.
- The map interface: improving the nearby clinic page to be more precise and practical.

REFERENCES

- [1] P. P. Company, “MyPurina – pet rewards,” App Store,
<https://apps.apple.com/us/app/mypurina-pet-rewards/id1658539065>. (accessed Jan. 21, 2024).
- [2] “Build apps for any screen,” Flutter, <https://flutter.dev/> (accessed Jan. 21, 2024).
- [3] M. Online, “5 แอพฯ ห้ามพลาด!! สำหรับคนรักสัตว์เลี้ยง,” MGR Online,
<https://mgronline.com/daily/detail/9580000014081> (accessed Jan. 21, 2024).
- [4] “Flutter architectural overview,” Flutter,
<https://docs.flutter.dev/resources/architectural-overview#:~:text=Flutter%20is%20designed%20a%20an,to%20be%20optional%20and%20replaceable>. (accessed Jan. 21, 2024).
- [5] “Pet icon pack,” Envato Elements, <https://elements.envato.com/pet-icon-pack-CCMBEQ5> (accessed Feb. 22, 2024).
- [6] K. T. Hanna and L. Rosencrance, “What is Google Firebase?,” Mobile Computing,
<https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase> (accessed Feb. 22, 2024).
- [7] “Firestore | Firebase,” Google, <https://firebase.google.com/docs/firestore> (accessed Feb. 22, 2024).
- [8] “Dart overview,” Dart,
<https://dart.dev/overview#:~:text=Dart%20is%20designed%20for%20a,forms%20the%20foundation%20of%20Flutter> (accessed Feb. 22, 2024).
- [9] Microsoft, “Visual studio code - code editing. redefined,” RSS, <https://code.visualstudio.com/> (accessed Feb. 22, 2024).

BIOGRAPHIES

NAME	Miss Russarin Eaimrittikrai
INSTITUTIONS ATTENDED	Saint Joseph Convent School, 2021: High School Diploma Mahidol University, 2025: Bachelor of Science (ICT)
NAME	Miss Supithcha Jongphoemwatthanaphon,
INSTITUTIONS ATTENDED	Satit Kaset Kamphaeng Saen School, 2021: High School Diploma Mahidol University, 2025: Bachelor of Science (ICT)
NAME	Miss Kanita Karunkittikun
INSTITUTIONS ATTENDED	Assumption Convent School, 2021: High School Diploma Mahidol University, 2025: Bachelor of Science (ICT)
NAME	Miss Nisakorn Ngaosri
INSTITUTIONS ATTENDED	Yothinburana School, 2021: High School Diploma Mahidol University, Present: Bachelor of Science (ICT)

LINK OF GITHUB OF THE PROJECT

<https://github.com/supithcha/HappyTails.git>