

Web-based Software Solution for Pet Kennel & Care Management (PawIn Software)

**S.R.C.N. Senanayake
2021**



Web-based Software Solution for Pet Kennel & Care Management (PawIn Software)

**A dissertation submitted for the Degree of Master of
Information Technology**

**S.R.C.N. Senanayake.
University of Colombo School of Computing
2021**



Declaration

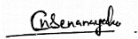
The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: S.R.C.N Senanayake

Registration Number: 2017/MIT/073

Index Number: 17550732



Signature:

Date: 29/11/2021

This is to certify that this thesis is based on the work of Ms. S.R.C.N Senanayake under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Mr. K.P.M.K. Silva



Signature:

Date: 30th Nov 2021

Abstract

Dogs and cats are playing a large role in human lives. People who are busy with their professions are seeking to provide those animals a good life. Pet owners always need to maintain good health of their beloved animals. However, finding good pet intuitions is a big problem. PawIn software will address this problem of pet owners by connecting pet institutes which are providing pet daycare, boarding, grooming and training services. This system will offer not only the mentioned services but also some veterinarian facilities and retail items via suppliers. People have the opportunity to buy dogs and cats via this application as well. Pet vaccinations, feeding and medication information will also be tracked through this software. Maintaining service dogs is also an important functionality of this application.

Acknowledgements

This would not have been successful without the support of many people. This is a good opportunity to give sincere thanks to the people who gave their maximum support to make this effort a success.

First I would like to express my sincere gratitude to University of Colombo School of Computing for offering this master's degree program to individuals like myself seeking knowledge in the IT industry, and also to all the staff members who guide students from the beginning of the program.

Secondly I would like to express my gratitude to Mr. K.P.M.K. Silva, my supervisor for the invaluable guidance, encouragement and precious support given throughout the journey. More importantly the valuable advice and criticisms were very fruitful.

Finally, I want to convey my heartfelt gratitude to my spouse, parents, and office colleagues who helped me succeed during the project by providing useful advice, direction, and support.

Table of Contents

Declaration.....	iii
Abstract.....	iv
Acknowledgements.....	v
List of Figures.....	viii
List of Tables.....	x
Chapter 1- Introduction.....	1
1.1 Project Overview.....	1
1.2 Motivation.....	1
1.3 Objectives.....	2
1.3 Scope of the Study.....	3
1.4 Structure of the Dissertation.....	4
Chapter 2- Background.....	5
2.1 Introduction.....	5
2.2 Requirement Analysis.....	5
2.3 Review of Similar Systems.....	8
2.3.1 Introduction to Kennel Management Software.....	8
2.3.2 Similar Systems in World Context.....	8
2.3.3 Similar Systems in Sri Lankan Context.....	12
2.4 Related Technologies.....	13
2.4.1 Angular 2+.....	13
2.4.2 Spring Boot.....	14
2.4.3 REST API.....	14
2.5 Related Design Strategies.....	15
2.5.1 Object- Oriented Design.....	15
Chapter 3- Design.....	17
3.1 Introduction.....	17
3.2 System Architecture.....	17
3.3 Proposed System Design.....	19
3.4 Database Design.....	20
3.5 Design Techniques.....	21
3.5.1 Class Diagram of the Proposed System.....	22
Chapter 4- Implementation.....	34
4.1 Introduction.....	34
4.2 Implementation Environment.....	34

4.2.1 Back End.....	34
4.2.2 Front End	34
4.3 Development tools and techniques	35
4.3.1 Tools	35
4.3.2 Technologies.....	35
4.4 Quality of the Solution.....	36
4.5 Main User Interfaces.....	37
4.5.1 Login Page.....	37
4.5.2 Create Appointment.....	38
Chapter 5-Testing and Evaluation	41
5.1 Introduction.....	41
5.2 Testing Procedure	41
5.2.1 Manual Testing	41
5.2.2 Test Automation	42
5.3 Test Plan and Test Cases.....	42
5.3 Sample Test Results.....	50
5.4 User Evaluation.....	59
Chapter 6- Conclusion	62
6.1 Introduction.....	62
6.2 Problems Encountered	62
6.3 Lessons Learnt	62
6.4 Future enhancements	63
References.....	64
Appendix A - Design Documentation.....	66
Appendix B-Management Reports	72
Appendix C – Code Listings.....	79
Appendix D - User Documentation	84
Appendix E- User Evaluation Results	93

List of Figures

Figure 2.1: Flow Chart of the Existing System.....	6
Figure 2.2: Data Flow Diagram of the Existing System.....	7
Figure 2.3: Live Sitter Map of the Existing System in Precise Petcare	10
Figure 2.4 : Branded Invoices and Quotes of the Existing System in Precise Petcare	11
Figure 3.1: Proposed system architecture of PawIn Software	18
Figure 3.2: ER Diagram of the Proposed System	21
Figure 3.3: Class Diagram of Proposed Solution.....	33
Figure 4.1: Login Page.....	38
Figure 4.2: Create Appointment Page.....	38
Figure 4.3: Appointment Details Page.....	39
Figure 4.4: Invoice Page	39
Figure 4.5: Cash Payment Pop up.....	40
Figure 4.6: Payment Successful Message.....	40
Figure 5.1: Validation of fields.....	51
Figure 5.2: Create Pet	51
Figure 5.3: View Pet	52
Figure 5.4: Add/view allergy details to particular Pet	52
Figure 5.5: Add/view medical and physical disabilities' details to particular Pet.....	53
Figure 5.6: Add/view injury details to particular Pet.....	53
Figure 5.7: Add/view medication details to particular Pet	54
Figure 5.8: Add dietary details to particular Pet.....	54
Figure 5.9: Create Appointment-Select Pets.....	55
Figure 5.10: Create Appointment-Enter Appointment details (Resource based)	55
Figure 5.11: Create Appointment-Invoice.....	56
Figure 5.12: Create Appointment-Pay by Cash	56
Figure 5.13: Create Appointment successfully	56
Figure 5.14: Create Appointment-Enter Appointment details (Employee based)	57
Figure 5.15: Add Resource.....	57
Figure 5.16: Add Employee	58
Figure 5.17: User Evaluation Form (Q1-Q2).....	59
Figure 5.18: User Evaluation Form (Q3-Q4).....	60
Figure 5.19: User Evaluation Form (Q5-Q6).....	60
Figure 5.20: User Evaluation Form (Q7-Q8).....	61
Figure 5.21: User Evaluation Form (Q9-Q10).....	61
Figure A.1: Use Case Diagram 1 of Proposed Solution	67
Figure A.2: Use Case Diagram 2 of Proposed Solution	57
Figure A.3: Use Case Diagram 3 of Proposed Solution	69
Figure A.4: Activity Diagram of Proposed Solution	71
Figure B.1: Feeding Report	73
Figure B.2: Feeding Report-Downloaded Report	73
Figure B.3: Medication Report	74

Figure B.4: Medication Report-Downloaded Report.....	74
Figure B.5: Check In Report.....	75
Figure B.6: Check In Report-Downloaded Report	75
Figure B.7: Check Out Report	76
Figure B.8: Check Out Report-Downloaded Report.....	76
Figure B.9: Revenue Report	77
Figure B.10: Revenue Report-Downloaded Report.....	77
Figure B.11: List of Service Dogs	78
Figure B.12: List of Service Dogs-Downloaded Report.....	78
Figure D.1: Login page	84
Figure D.2: Display login user privilege.....	85
Figure D.3: Navigation bar	85
Figure D.4: Common buttons	86
Figure D.5: Date picker using system.....	86
Figure D.6: Time picker using system.....	86
Figure D.7: Common grid view used in modules.....	87
Figure D.8: Display error messages.....	87
Figure D.9: Save message display	88
Figure D.10: Create Customer Profile	89
Figure D.11: Add Pet	89
Figure D.12: Add Medication/Add Dietary/Add Vaccination by selecting pets	90
Figure D.13: Company Portal	90
Figure D.14: Pet Live Board.....	91
Figure D.15: Search Employee	91
Figure D.16: Add Schedule for selected employee	92
Figure D.17: View employee schedule for selected employee.....	92
Figure E.1: Responses for Software Recommendation	93
Figure E.2: Responses for Software’s Ease of Use.....	94
Figure E.3: Responses for Features of Software.....	94

List of Tables

Table 2.1: Concepts of Object Oriented Design	16
Table 5.1: Test Case-Verify Login functionality	42
Table 5.2: Test Case-Create Profile	43
Table 5.3: Test Case- Edit Profile.....	43
Table 5.4: Test Case- Create Appointment.....	44
Table 5.5: Test Case- Checked In /Checked Out Pet.....	45
Table 5.6: Test Case- Retail.....	45
Table 5.7: Test Case- Suppliers	45
Table 5.8: Test Case- Pet Live Board-Check in Pet for current date.....	46
Table 5.9: Test Case- Add/Edit Employee	46
Table 5.10: Test Case- Employee schedule	46
Table 5.11: Test Case- Add/Edit Veterinarian.....	47
Table 5.12: Test Case- Add/Edit/Delete Service Dogs.....	47
Table 5.13: Test Case- Feeding Report	48
Table 5.14: Test Case- Medication Report	48
Table 5.15: Test Case- Check In Report	49
Table 5.16: Test Case- Check Out Report	49
Table 5.17: Test Case- Revenue Report	50
Table 5.18: Test Case- List of Service Dogs	50
Table A.1: Use Case 1	66
Table A.2: Use Case 2	68
Table A.3: Use Case 3	70

Chapter 1- Introduction

1.1 Project Overview

Software resolutions can be used to handle real world problems in an innovative way. For practically all applications, software offers cost effective, accurate, user friendly and reusable solution. Programming, databases, web and multimedia have been the most extensively utilized for development of application software. Variety of software engineering tools are available for software applications at present. This document elaborates a project to create a software solution for the domain of pet kennel management including appointment creations, veterinarian services and pet shop handling. In doing so the introduction outline motivation, objectives, background of the study, scope for the underline project, feasibility study, requirement analysis and current progress of the project.

1.2 Motivation

Working people choose to give a nice life for animals because they play such an important role in our lives. Dogs and cats are the more familiar to human when comparing to other animals. Pet owners are facing difficulties when they are out of their residence for whole day or couple of days. It is required to look after those pets by providing daycare, boarding, grooming and training services. While offering the mentioned services to pets it will be grateful to maintain a good health via the same application.

There are people who wish to buy a pet or pet related products with a trust as well. On the other hand, dogs could be used to direct the blind, disabled or old people after a successful training session. There are many systems which are allowing for veterinary services as well as pet selling in Sri Lanka. Pet in house care or kennel management is ongoing developing area in the country and currently most of these services are provided manually. Software which would allow pet in-house care including the veterinary services, maintaining pet shops via suppliers, providing service dogs is also lacking for the country. It is demanding to introduce a web-based system for pet in-house care management because it would reduce labor time, rate of error and enhance productivity and the quality of pet care.

1.3 Objectives

Aim of the project is to develop the web-based application for pet care institutions and pet owners in order to book appointments depending on their requirement while providing all pet facilities including feeding, medications, vaccinations, grooming etc.

Objectives can be described as follows:

- To help people with busy schedules to look after their pets. This would be supportive when pet owners are not in their residence for some days. They can trust a kennel management software to look after the pets.
- To prevent diseases and maintain good health of dogs and cats. This upcoming software would collaborative with veterinaries and track the behaviors of the in-house pets in order to identify their health individually.
- To send vaccination reminders for the pet owners that would remind the expiration dates of required pet vaccinations. Mandatory vaccinations could be injected via this application and pet owners would know about the time that need to get the vaccine for their pets hence they would receive an email reminder.
- To provide attractive platform for pet suppliers to offer pet products via proposed system. Suppliers who wish to provide pet related products such as pet collars, foods and chains can access this upcoming system.
- To expand opportunities for veterinarians in order to make consultations of pets. Veterinarians would have the ability to connect to the software and examine pets who are not well and it would be a good opportunity to expand their skills while having an income.
- To support blind, disabled or elders for the direction. Dogs can be trained to assist people who are physically inactivate. These training sessions would also maintain through this proposed system.
- To analyze pet feeding, medication, check in and check out daily or monthly basis. Customized reports would be generated based on required time period and these reports could be used for analytical purposes.

1.3 Scope of the Study

Proposed system should be able to add clients by entering their main details including first name, last name, email, phone number and pet details. Clients can have single or multiple pets and pet name, breed, gender, weight, birth date would be maintained via customer profiles. Pet owners should be able to book appointments by selecting services such as Daycare, Boarding, Grooming and Training and selecting dates and times according to their requirement. System would inform to pet owner if the selected date or time is not available and then owner can select any other available date or time. System should be able to monitor ongoing pet activities through live broadcast for particular pet owners. Pets who are checked in for the current date would be tracked via this board and pet information such as name, feeding and medication would be displayed.

Pets and pet related retails can be sold via this upcoming software and it should be able to handle suppliers for pet selling and maintain pet shops. Customers can buy pets or pet related products by connecting to the system. Customers should be able to do payments of services by the use of cash, credit or pay at pick up options. System should be able to provide service dogs in order to direct disabled or people who are in their old ages. Administrator should be able to checked in and checked out pets by selecting dates and times. Checkout would be finalized after successful payment has been made by the particular pet owner. System should have two categories of services namely resource based and employee based. Daycare and boarding services are categorized as resource based services while grooming and training services categorized as employee based services.

System should be able to perform resource (rooms) handling based on breed and size of the pet specifically for resource based services. System should be able to maintain veterinary for required pets who are not well with behavior or injured. This application will provide pet vaccinations including Rabies, DHLPP, Bordetella, Parvo and Lepto and it should also maintain feeding and medication of pets. System should be able to send emails and text messages for vaccination reminder, expired vaccinations and birthday reminders of pets. Employee Schedules for employee based services would be maintained by the upcoming system through updating time slots of employees. System should be able to generate reports based on checked in, checked out, feeding, medication in order to analyze the pets' behavior for desired date or date range.

1.4 Structure of the Dissertation

In relation to the dissertation, this is an attempt to present the overall perspective of the project. As a result, these project-related details are broken down into five chapters to make the process of comprehension easier. Introduction, Background, Methodology, Evaluation and Conclusion are the chapters that are included in the dissertation.

Chapter 2- Background

2.1 Introduction

In this chapter, similar project research areas are explored with the hopes of eliciting and assessing the systems' underpinning implementation and technological principles. Furthermore, the literature review is conducted to view the problem studied and proposed solution generally with different perception that aid to build up a quality solution.

2.2 Requirement Analysis

The process of the existing system would be depicted using diagrammatical representations in this section. Figure 2.1 defines the flow chart of the basic feature of existing system. Figure 2.2 describes the Data Flow Diagram of existing system by considering both computerized and manual based functionalities.

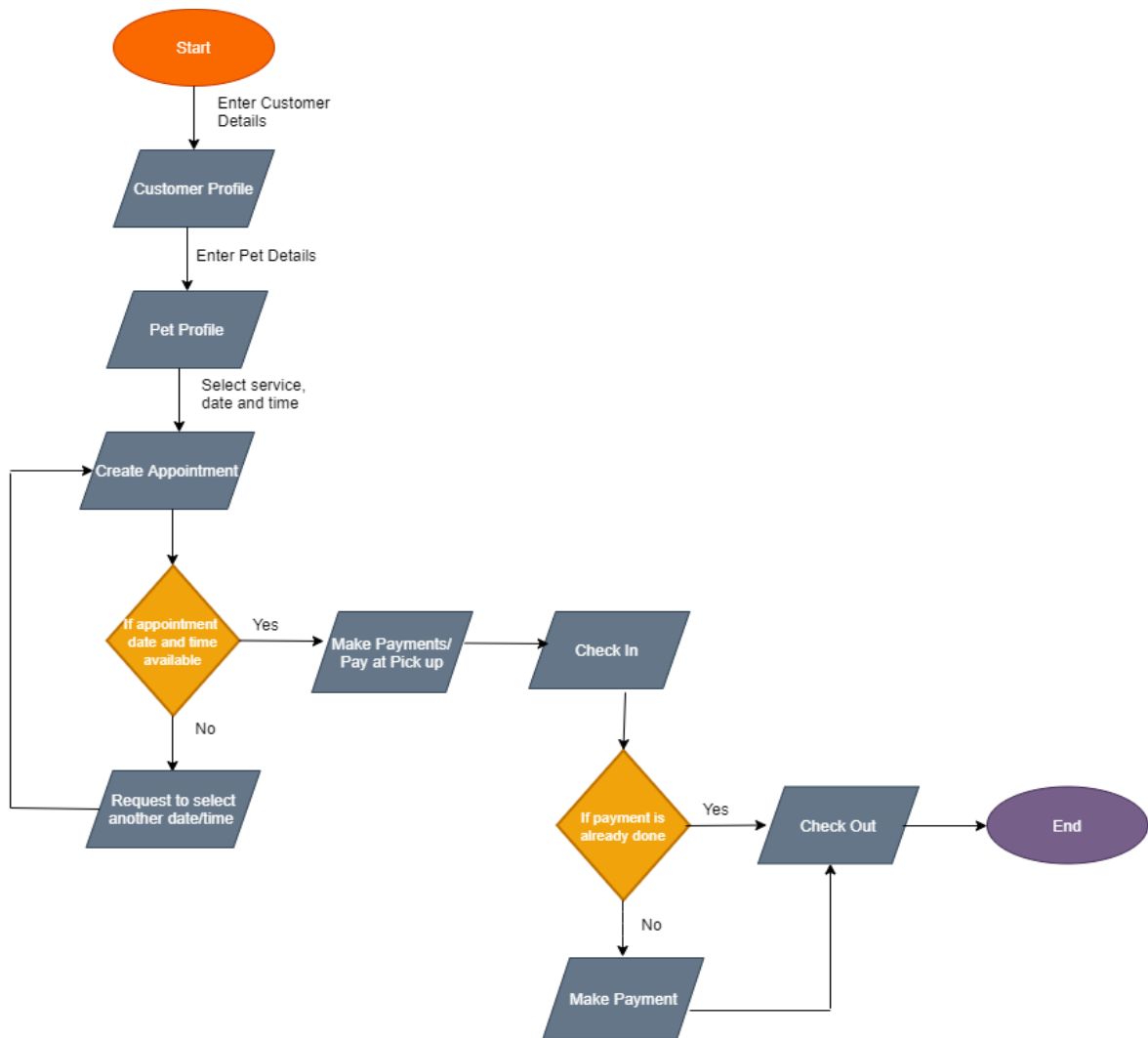


Figure 2.1: Flow Chart of the Existing System

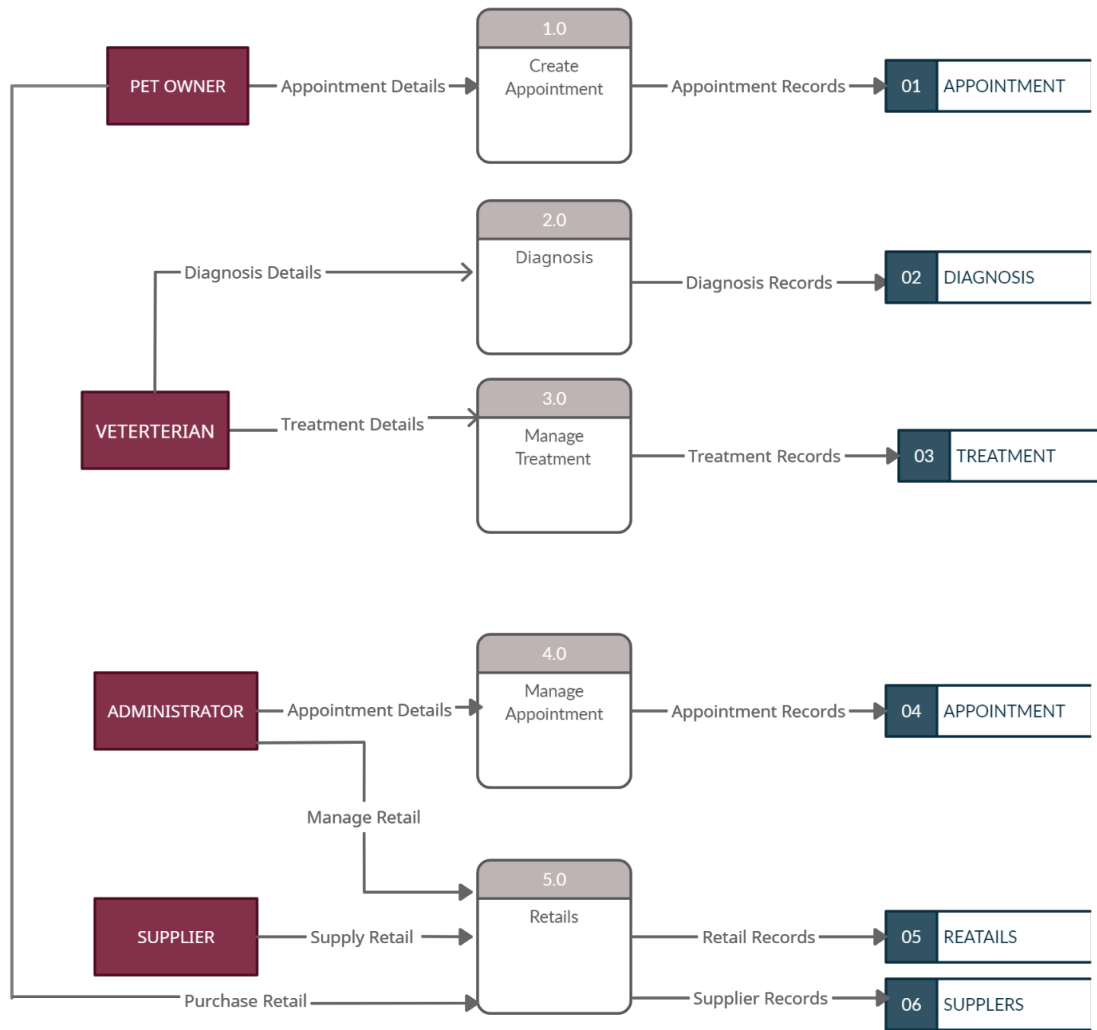


Figure 2.2: Data Flow Diagram of the Existing System

2.3 Review of Similar Systems

In this section similar systems in global context as well as local context would be deliberated in detail.

2.3.1 Introduction to Kennel Management Software

Kennel management software is based on pet care facility management that permits employees, pet daycare, boarding and training institutes to handle various administrative tasks including client management, appointment scheduling and more. Authorities can utilize the dashboard to keep track of pet check-in/out times, see upcoming bookings on a calendar, and visualize statistical data with the aid of graphs and charts.

Pet groomers can use this application to book appointments for several pets with the same owner, keep track of their medication, feeding, and exercise schedules, and record client information in a database. Managers can control employee access, choose preferred payment methods, and produce commission reports. Reminders, discount management, a self-service portal, online booking, inventory management, and barcoding are among the other features.

2.3.2 Similar Systems in World Context

2.3.2.1 Case Study: Revelation pets

Revelation pets (Revelationpets, 2021) is a dog day-care, cattery and kennel management software in global context which allows to online booking, client and pet management, pet service management, calendar management and pet birthday tracking. Unlimited, comprehensive client and pet profiles are allowed in this software while string vaccination records, tracking expiration-dates and sending vaccine reminder alerts. This also manages reservations, appointments and payments. This exists a mobile-friendly layout. Customers have the ability to verify facility's availability. They have the option to self-register and upload their own information. Customers can also view their pet activity during their pets' stay. Instant overview of arrivals, departures, single-click check in or out feature are managed through this system.

This application accommodates changing different rates for sharing pets, additional charges or upsell products under reservation and service settings, automatic email and SMS booking, customize email text for all client communication, and a flexible special rate solution to quickly amend rates and apply discounts under reservation and service settings. Customers can purchase packages that include a number of services that are complementary. Individual employee calendars would be used with 'unavailable' periods blocked off. It also syncs with Google Calendar and Google Maps, allowing customer to view all of his appointments by day or week, as well as by service, such as grooming, training or walking.

2.3.2.2 Case Study: Precise Petcare

Precise Petcare (Petcare, 2021) is one of the software solutions that manages every aspect of pet sitting business including automated scheduling, branded invoices and quotes, reminder emails, employee management and pet profiles and is fully customizable to specific needs. A schedule is running for everyone including managers, sitters and clients. The Live Sitter Map, accessible from the main dashboard, displays each sitter's most recent recorded location. Customers will see a designated pin box over each sitter's location, which they can click to get more information about, such as visit status and duration. It would automate the process of finding out as soon as possible if a sitter is late for a visit (or does not arrive at all). Late Arrival Alerts are triggered by this system when a sitter does not come on time. It's normal for busy pet owners to establish a routine with their sitters. Pet owners save time because staff does not have to schedule these visits manually. Their ongoing schedule look ahead feature enables for fully automatic scheduling of recurring visits. The system will check overbooking conflicts based on the estimated length set for each service and alert the customer when one exists.

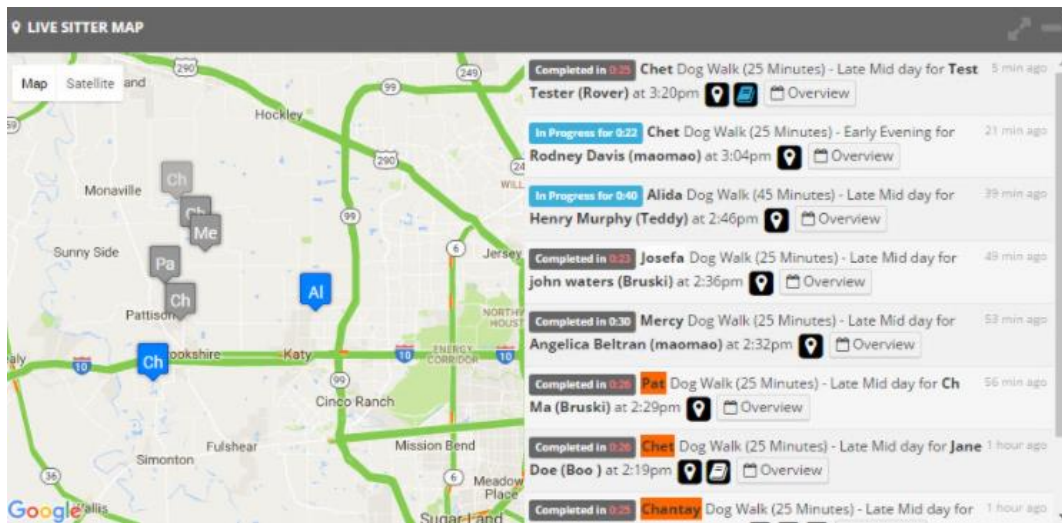


Figure 2.3: Live Sitter Map of the Existing System in Precise Petcare
 Source: (Petcare, 2021)

This app provides iCal feeds for any appointment in the system for anyone whose lives revolve around a specific calendar. This implies that employees and clients may access their schedules through Google Calendar, Mac Calendar, Outlook, and other calendar apps (Petcare, 2021). All customer bills may be made, kept, and tracked under the billing area, which also includes an option for auto invoice generation, which allows you to set a schedule for the system to generate invoices. To gain significant insight into pet institutions, a variety of income and employee compensation statistics or charts can be prepared. Assignment reports, which show when income was completed, and billing reports, which are best used for cash flow and tax purposes, are examples of reports that provide crucial information. Surcharges are a strong tool to add additional costs or pay for visits based on pre-determined circumstances. Surcharges for holidays, weekends, and specific times of day, places, and last-minute requests are all common. Custom surcharges can also be created by pet intuitions.

This solution keeps clients informed about their visits without requiring them to log in by sending out automatic email confirmations or manually sending them on demand. The communication center panel displays all client and staff messages in a "snapshot" on the dashboard, allowing pet intuitions to quickly identify the most appropriate comments for each customer. Closing a client's communication view will remove them from a panel until a new communication is received. It's critical that clients have a quick and easy option to get time-sensitive data. Customer Timeline on

their dashboard shows a chronological list of pet care journals and invoices, allowing clients to see all of their communication, images, and transactions in one place (Petcare, 2021).

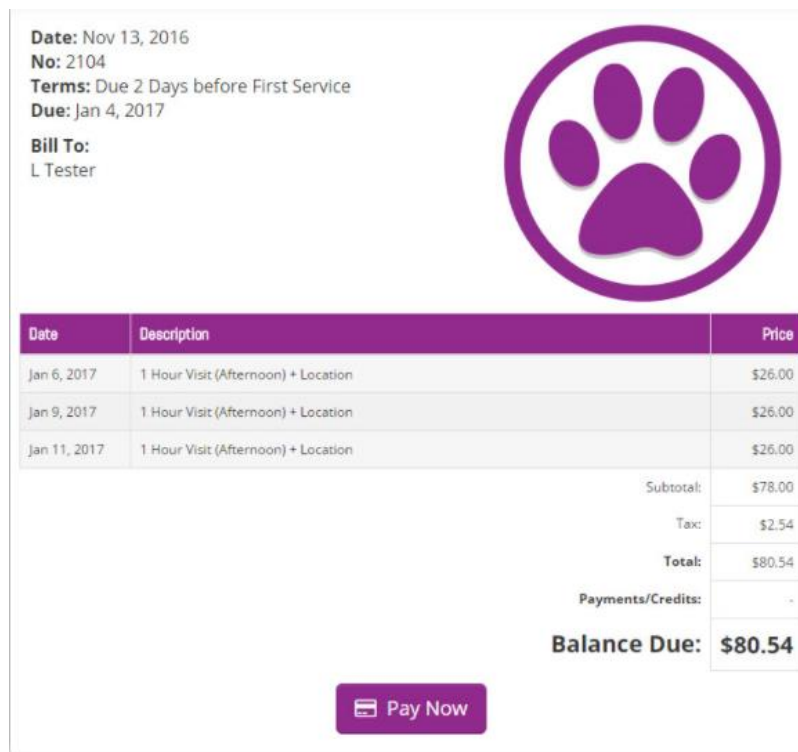


Figure 2.4 : Branded Invoices and Quotes of the Existing System in Precise Petcare
Source: (Petcare, 2021)

2.3.2.3 Case Study: PetExec

PetExec (Petexec.net. 2021) is a cloud-based pet care business management solution for busy pet professionals to manage daycare, boarding, grooming, training and retail related to pets. In order to update pet care business management services, PetExec includes comprehensive scheduling tools, automatic appointment reminders and a multitude of reports. PetExec incorporates many features which are intuitive to staff and are important tools to keep and electronic record of all guests, histories, records, incidents and other important details. Pet owners can make safe credit card payments, book appointments in real time, and check their pets' records.

With a dedicated application for iOS and Android alike, clients will be able to schedule themselves for all of pet institutions' core services, upload files to their account, purchase packages, and etc. Paws On Board is PetExec's on-boarding program it covers a set up discussion and initial set up of basic products and services by one of their knowledgeable coaches. Paws On Board also includes an initial web training (hour and a half webinar) and this session is recorded for customer's review and future staff training. Automated text message appointment reminders are available in this system and it also sends reminder messages in text from boarding, grooming, scheduled services appointments and group training classes.

2.3.3 Similar Systems in Sri Lankan Context

2.3.3.1 Case Study: Bark street

This is kind of a website which offers professional services to all pet owners using manual based approach. This application maintains an online store in order to sell dog related products to customers. They provide several services which include show handling and conditioning amongst the specialized offerings (Barkstreet.lk, 2021). This includes their own obedience training school which pets can be brought to or if customer prefers, their trainers will be available to visit customer's home at a convenient time to train particular pet. Barkstreet is a pet-friendly boarding facility that offers excellent service. Each of their boarding kennels has been converted into a suite for certain pets. Customers can leave their pets for a day, a weekend, or longer. They will feel totally at home with their pet, and they will ensure that they receive better treatment.

This website also offers consultations for new pet owners who are unclear what to feed or how to care for their animals, or who have a problematic dog they are unsure how to handle. They would guide the pet owners to find the right puppy according to their requirement and also will direct how to feed pets creating a nutrition plan or the vitamins that need to be given to the pets. Grooming is required for all dogs, regardless of coat type, and they offer this service. They will groom the pet using the appropriate equipment depending on the type of coat it possesses. Grooming is more than just brushing the dog; it also includes pet care.

2.3.3.2. Case Study: Best Care

Best Care (Bestcare.lk, 2018) is an animal hospital which allows to book appointments for veterinary services via website while providing inward care and boarding, pet taxi services, mobile veterinary services, pet shops and pharmacy as walk in services without using a software.

2.4 Related Technologies

This section illustrates the related technologies with respect to Kennel Management Software.

2.4.1 Angular 2+

Instead of JavaScript, this is an open-source solution based on TypeScript (angular.io, 2021). Angular 2 is the generic title for all Angular versions with others AngularJS. Angular 2, 4, 5-6, 7, 8, 9, 10 are the current versions. Single-page websites, corporate websites, e-commerce websites, social networks, and complicated mobile apps are just a few of the tasks that Angular 2 can handle. In compared to other JS frameworks and Angular 1, the functionality of Angular 2+ is superior. It enables developers to determine that Angular 2+ is suitable for all types of projects. Whatever project you're working on, Angular 2+ will provide dependable support.

Angular 2+ offers great functionality for its uses. Data that needs to be presented in particular application can be easily transferred. There is no need to use any third-party libraries to develop basic functionality for your application because the Angular team has given an official library. This would result in improved security and code quality. It enables the creation of feature-rich and responsive apps. Angular is a full-featured web design framework for creating responsive websites. One of the most useful features is that there is just one recommended method for creating a component, service, or module. In practice, it ensures consistency across the code base and prevents other developers from assuming on how something was developed.

2.4.2 Spring Boot

Spring Boot is a framework created by Spring Team to make launching and developing new Spring Applications easier (spring.io, 2021). It provides defaults for code and annotation configuration so that new spring projects may be started quickly. It uses an approach called "Opinionated Defaults Setup" to eliminate a lot of boilerplate code and configuration in order to simplify the development, unit test, and integration test processes. Spring Boot makes it simple to create Spring-based Java programs, which is how the proposed system will be created. It saves a lot of time during development and boosts productivity by reducing the need to write a lot of boilerplate code, annotations, and XML settings. Spring Boot applications may be easily connected with the Spring Ecosystem, which includes Spring JDBC, Spring ORM, Spring Data, Spring Security, and more. It has embedded HTTP servers such as Tomcat, Jetty, and others that make developing and testing web applications a breeze.

2.4.3 REST API

REST (Representational State Transfer) is any interface between systems that uses HTTP to get data and perform operations on it in any format, including XML and JSON. This is becoming a more common alternative to other standard data interchange protocols like SOAP (Simple Object Access Protocol), which have a lot of capacity but are also somewhat complicated. Using a simpler data-processing approach, such as REST, is sometimes preferable. The REST protocol completely separates the server and data storage from the user interface. When it comes to development, this has some advantages. It improves the interface's portability to other types of platforms, promotes project scalability, and allows distinct components of the development to expand independently, for example.

REST's separation of client and server has one obvious advantage. Developer can scale the product without too much difficulty. They can move to different servers or update the database in any way they want as long as the data from each request is sent appropriately. The split makes it easy to have the front and backend servers on separate servers, making the apps more versatile to use. The REST API is invariably independent of platform or language. It adjusts to the type of syntax or platform being used at all times, giving developers a lot of flexibility when changing or testing new contexts. Developers can use Java or Node.js servers with a REST API. The only requirement is that responses to queries must always be made in the language used for information exchange, which is usually XML or JSON.

2.5 Related Design Strategies

This section discuss about design strategy related to Kennel Management software which is a web based application.

2.5.1 Object- Oriented Design

The software will be created using an object-oriented design method, which means that everything will be treated as an object. As a result, each item has its own set of actions and behaviors. This is distinguished by classes, because all objects belonging to the same class will exhibit identical behavior and perform comparable functions. In some way, each thing will be distinct from the others. This is determined when the object is created and initialized with some values that provide the object its distinct behavior.

The object-oriented design approach is preferred over the function-oriented design approach because real-world entities may be easily implemented in the computer world with the object-oriented approach. This method can be used to implement certain very basic object characteristics like polymorphism, inheritance, abstraction, and encapsulation.

Following table 2.1 illustrates important concepts of Object Oriented Design.

OOD Concept	Description
Objects	Objects refer to all entities involved in the solution design. Pet owner, pet, supplier, veterinarian, employee are treated as objects. Every entity has some attributes associated with it, as well as methods for manipulating those attributes.
Classes	A class is a way of describing an object in a more broad way. A class's instance is an object. The class defines all of the characteristics that an object can have, as well as the methods that determine the object's functionality. Attributes are retained as variables in the solution architecture, and functions are described using methods or procedures.
Encapsulation	Encapsulation is the organization of characteristics (data variables) and methods (data operations) together. Encapsulation not only groups together critical information about an object, but it also prevents data and operations from being accessed from the outside world. This is known as information hiding.
Inheritance	OOD allows similar classes to be stacked in a hierarchical fashion, with lower or sub-classes being able to import, implement, and reuse variables and methods from their immediate super classes. This property is known as inheritance. This makes expressing specified classes and creating generalized classes from specific ones easily.
Polymorphism	Methods performing identical duties but differing in arguments can be given the same name in OOD languages. This is defined as polymorphism, because it allows a single interface to perform functions for a variety of types. A piece of the code is performed depending on how the function is called.

Table 2.1: Concepts of Object Oriented Design

Chapter 3- Design

3.1 Introduction

The software design phase is an iterative process in which the requirements acquired during the analysis are turned into a "blueprint" for the system's development. Following that, detailed functional and behavioral requirements can be developed. This chapter will describe the system architecture in detail, design techniques including database design and behavior of the proposed system in diagrammatical representation.

3.2 System Architecture

Micro services architecture would be used to develop this proposed solution. Micro services is one of the architectural patterns which has emerged from the world of domain-driven design, continuous delivery, platform and infrastructure automation, scalable systems and persistence (microservices.io. 2021). A micro services architecture takes the approach of “gather together those things that change for the same reason, and separate those things that change for different reasons.” It also includes loosely coupled services that may be developed, deployed, and maintained on their own. Each of these services is in charge of a distinct activity and can interface with other services via simple APIs in order to solve a larger, more complex business challenge.

Benefits of Micro services architecture can be defined as follows:

- Highly maintainable, testable
- Loosely coupled
- Self-deployable
- Organized around business capabilities
- Owned by a small team

Following figure 3.1 illustrates the proposed system architecture of PawIn Software.

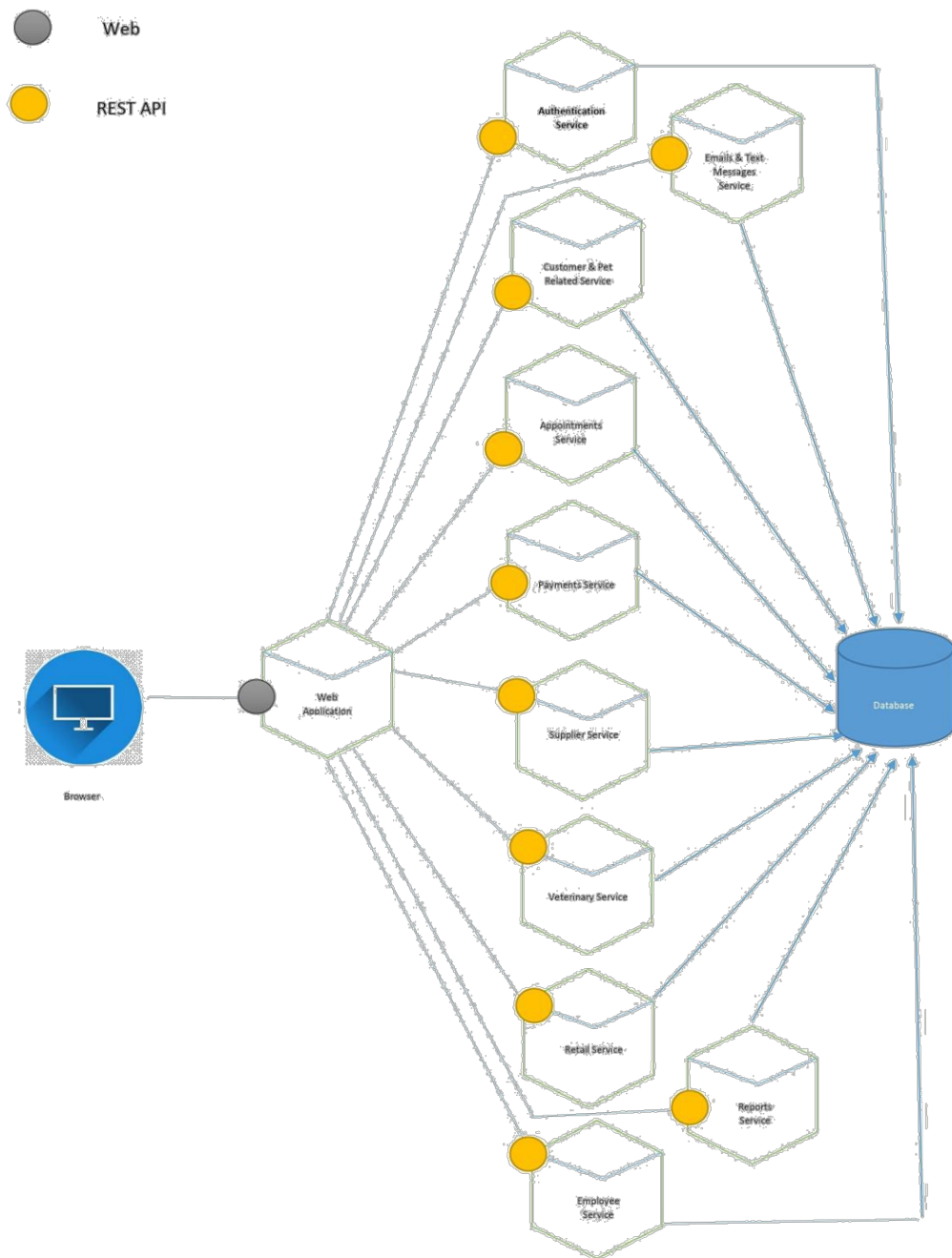


Figure 3.1: Proposed system architecture of PawIn Software

Web Application can be accessed via a web browser. Services can be written for authentication, customer and pet related functions, appointments, payments, supplier, veterinary, retail, employee, emails and text messages and reports. These services will be developed using REST API. All the services will be directed to central database.

3.3 Proposed System Design

By eliminating manual human effort and generating reports to make timely decisions, the system will encourage users to carry out their obligations successfully and efficiently. To make the creation and comprehension of the proposed solution easier, it has been divided into a few components. The modules are listed below.

- Authentication module

User management for different access levels via permitting login, logout functions and user registration will be management through this module

- Customer and Pet module

Customer and pet profile creation, managing vaccinations, feeding and medication details and handling service dogs will be covered over this module

- Appointments module

Resource based services such as daycare and boarding and also employee based appointments namely grooming and training are allowed to select as appointments. Pet check in and check out functions are also handled via this module.

- Payments module

Cash and pay at pick up options will be allowed for pet owners and it will be controlled by creating invoices via this module.

- Supplier management module

Suppliers who are providing retail items and pets for shopping purposes will be management using this module.

- Veterinarian management module

Veterinarians who are offering pet medical services will be managed via this module.

- Retail module

Pet retail items such as food, collars, belts and also pets for the purpose of phrasing will be managed via this. Retail inventory count will be captured using this module.

- Employee management module

Employees who are offering employee based services and their schedules will be monitored in this module

- Emails and text messages

Pet birthday wishes, vaccination reminder emails for expired vaccinations as well as vaccinations which will be expired within 10 days will be managed in this module.

- Reports module

Customized reports such as feeding, medication, check in and check out of pet will be covered in this module.

3.4 Database Design

Data modeling is the last step in the database design process. The structure of business objects utilized in the system is defined by database design. To prevent redundancy, the database tables were normalized to the third normal form. Figure 3.2 depicts the ER diagram of the proposed system.

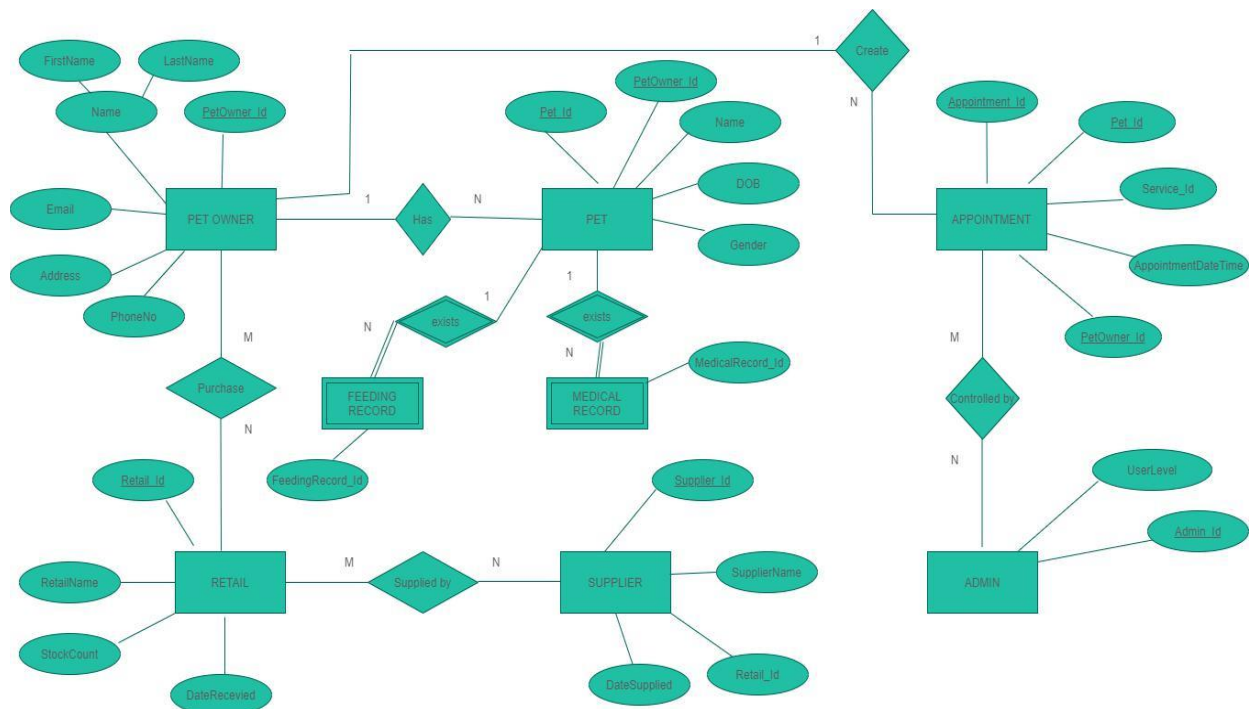


Figure 3.2: ER Diagram of the Proposed System

3.5 Design Techniques

The approaches used to represent the system are known as design techniques. Modern structured design, rapid application development, and prototyping are examples of design methodologies. Object-oriented design was chosen above the others because it allows for code reuse and recycling, as well as design benefits and maintainability through the use of objects and classes (Smashing Magazine. 2021).

In Object Oriented Design, the Unified Modelling Language (UML) is very important. UML allows programmers to readily grasp object models, making it easier for them to create software. The following are some of the structural and behavioral UML diagrams that are used in the design of proposed systems.

- Class diagrams – depicts the required objects and their relationships.
- Use-Case diagrams with narratives — demonstrates what the system should perform. Object states at a certain time are depicted in activity diagrams.

The language utilized to construct the system is Java, which has a wide range of object-oriented characteristics. The system was designed utilizing Micro services architecture to simplify and strengthen the development context.

3.5.1 Class Diagram of the Proposed System

Class diagrams aid in the conceptual and domain modeling processes. A conceptual model illustrates objects and classes from the viewpoint of a certain system user. Figure 3.3 defines the class diagram of the proposed solution.

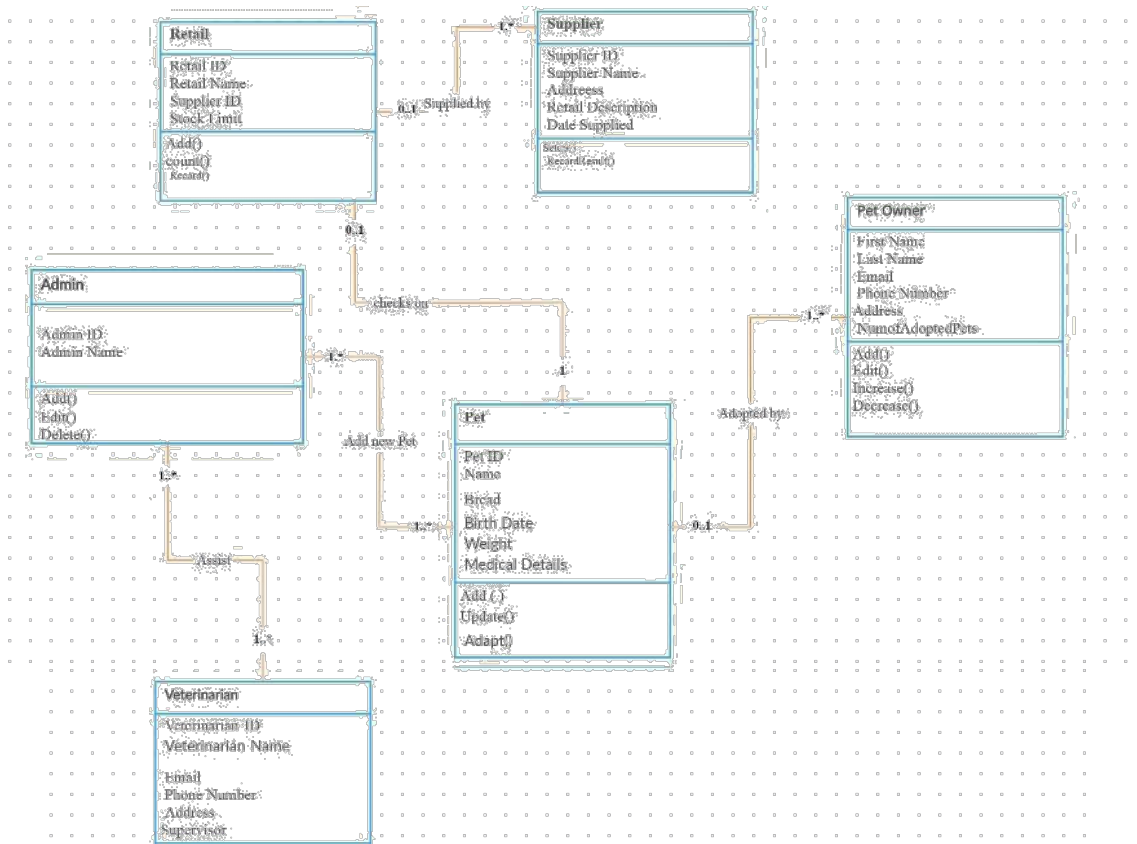


Figure 3.3: Class Diagram of Proposed Solution

Chapter 4- Implementation

4.1 Introduction

The implementation phase converts system specifications into executable code. Convert the design into user-friendly views with front end interfaces and back-end systematic logics for performing functionalities in this step. During the coding and development phase, a pleasant language and appropriate tools are defined.

4.2 Implementation Environment

Back end and front end applications are considered in implementation process. Software requirements that are essential to develop these two applications are described below.

4.2.1 Back End

Back end of the system is developed using spring boot framework. An embedded tomcat server has contained in the framework and .jar file which is in the target of the project will be run in order to get the access for the back end of the system. REST API will be used to write the web application in API level.

4.2.2 Front End

Angular is used as the front end of the system. A set of files will be created after built the Angular project. nginx is used as the front end server and it will be performed as a container. Angular and spring boot will be connected via reverse proxy. When user accesses the front end Angular application bearer token will be sent to reverse proxy. Reverse proxy consists of API header, body, Secured Socket Layer (SSL), redirection path and etc. Reverse proxy is coming with nginx. Request will be sent to backend sprint boot application via reverse proxy. Spring boot is connected with the database of the application. After retrieved data from database, it will be connected to the spring boot application again. Then it will be moved to angular application via reverse proxy. Angular application will show the content as user interface to the particular user.

4.3 Development tools and techniques

The tools and technology utilized to develop the system are listed below.

4.3.1 Tools

- IntelliJ IDEA

IntelliJ IDEA Ultimate edition is used as the IDE for this project. It increases developer productivity, improves developer focus, and reduces technical debt while also improving testing and debugging productivity, as well as improving testing strategies and reducing code maintenance work.

- My SQL Workbench

MySQL Workbench is used for the database creation. It offers a complete and simple solution for converting tables, objects, and data from Microsoft SQL Server, Microsoft Access, Sybase ASE, PostgreSQL, and other RDBMS to MySQL. On both Windows and other platforms, developers may convert existing programs to run on MySQL quickly and efficiently.

4.3.2 Technologies

- Java as development Language

Java is used to build applications that run on spring boot framework and programming language is a fully object oriented language.

- Spring Boot Framework

Spring Boot offers a flexible approach to setup Java Beans, XML configurations, and Database Transactions, as well as robust batch processing and management of REST endpoints. Everything is auto-configured in Spring Boot; no human configuration is required. It has a spring application

that is annotation-based. It contains an Embedded Servlet Container and simplifies dependency management.

- Angular

Angular is an HTML and TypeScript-based platform and framework for creating single-page client applications. TypeScript is used to write Angular. It provides a set of TypeScript libraries that may be used to construct the application's core and optional functionality.

Code segment in Appendix C illustrates a sample for this architecture in the system.

4.4 Quality of the Solution

This section describes the quality of the PawIn software with respect to below characteristics:

- Reliability

This application will be worked in different environments with different conditions. System will be tested as unit testing, integration testing and system testing using prepared test cases for each possible scenarios. It will direct the user to trust the system.

- Maintainability

Since the developer uses the object oriented concepts for the development, it will easy to add new code to the existing code. It will provide the ability to upgrade for new features, new technologies from time to time. The solution will be code effective because most of the tools that are used for the development are open source.

- Usability

The interfaces of the system is very user friendly and will allow users for simple navigation to achieve their destinations. New users have the ability to learn the system easily because of the simple navigations.

- Correctness

Main functionalities of the system will be covered by considering the different users of the system such as pet owners, suppliers, admin and veterinarians. All the calculations related to appointment service prices will carefully managed.

- Efficiency

This software will increase the efficiency of the current manual based system for pet kennel management. Time requirement complete a specific task would be more effective when using a software when it's compared to a task completed by a human being.

- Security

The system will provide high security by offering different access levels for different users. Unauthorized access and information loss will be prevented through this secured system. Jason Web Token (JWT) is used for tokenization and it's more secured mechanism rather than using only simple username and passwords. Bearer token is used at first in order to get a separate token for each session.

- Compatibility

PawIn software will work in desktop, laptop, tablet and mobile. It would be easy for the user to use the application in all devices according to the desire of him.

4.5 Main User Interfaces

In order to demonstrate system interface structure, the following section of the document simply shows a few important user interfaces.

4.5.1 Login Page

This screen (Figure 4.1) allows the user to log in with the correct user name and password to access the system.

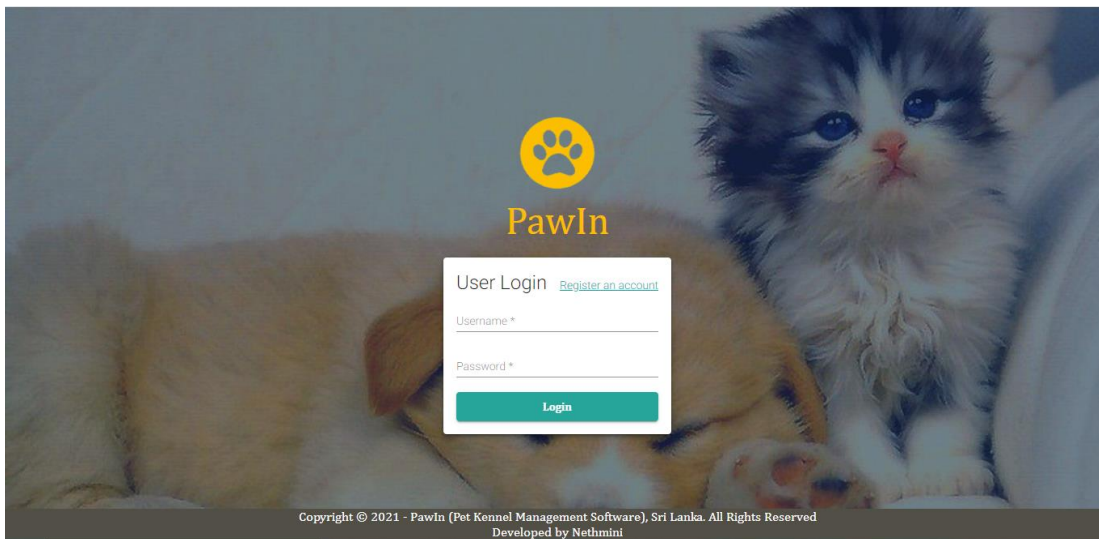


Figure 4.1: Login Page

4.5.2 Create Appointment

As indicated in figure 4.2, customer or the operator of the system needs select the pets for the particular appointment.

Create Appointment		
Name	Image	Select
Pinky		<input checked="" type="checkbox"/>
White		<input type="checkbox"/>
Nicky		<input type="checkbox"/>
nnjn		<input type="checkbox"/>

[Next](#)

Figure 4.2: Create Appointment Page

As the 2nd step of appointment creation, customer or the operator of the system has to enter appointment details as shown in figure 4.3.

Appointment Details

Service Category:

Service:

Check In Date:

Check In Time:

Check Out Date:

Check Out Time:

Find Room:

Figure 4.3: Appointment Details Page

As the 3rd step of appointment creation, customer or the operator of the system can see the invoice details as displayed in figure 4.4.

Invoice

Date	Service	Price(No. of Pets * No. of Days * Price)
2021-07-18 ->> 2021-07-18	Full day	2 * 1 * 1000

Total(Rs.) 2000

Figure 4.4: Invoice Page

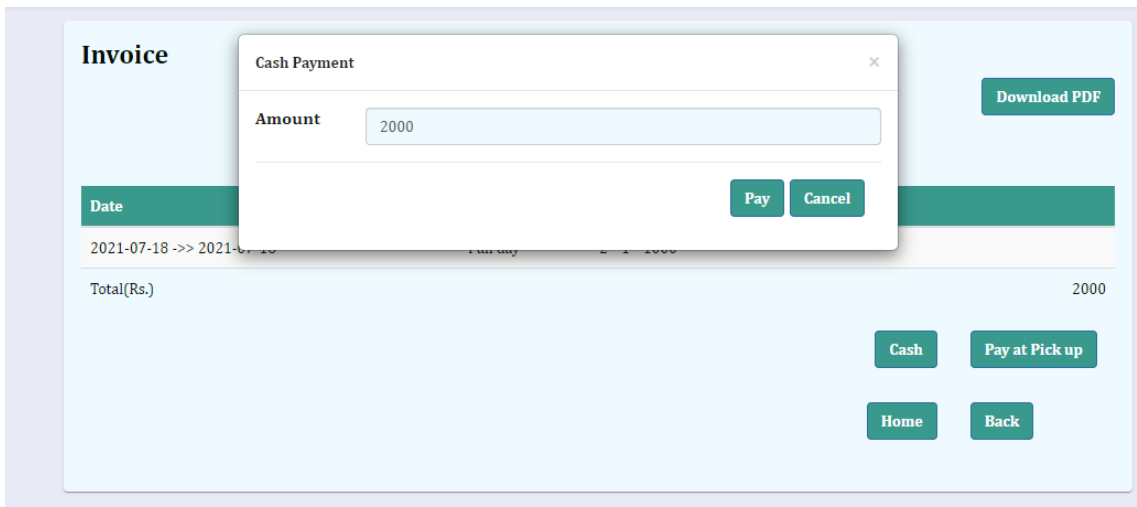


Figure 4.5: Cash Payment Pop up

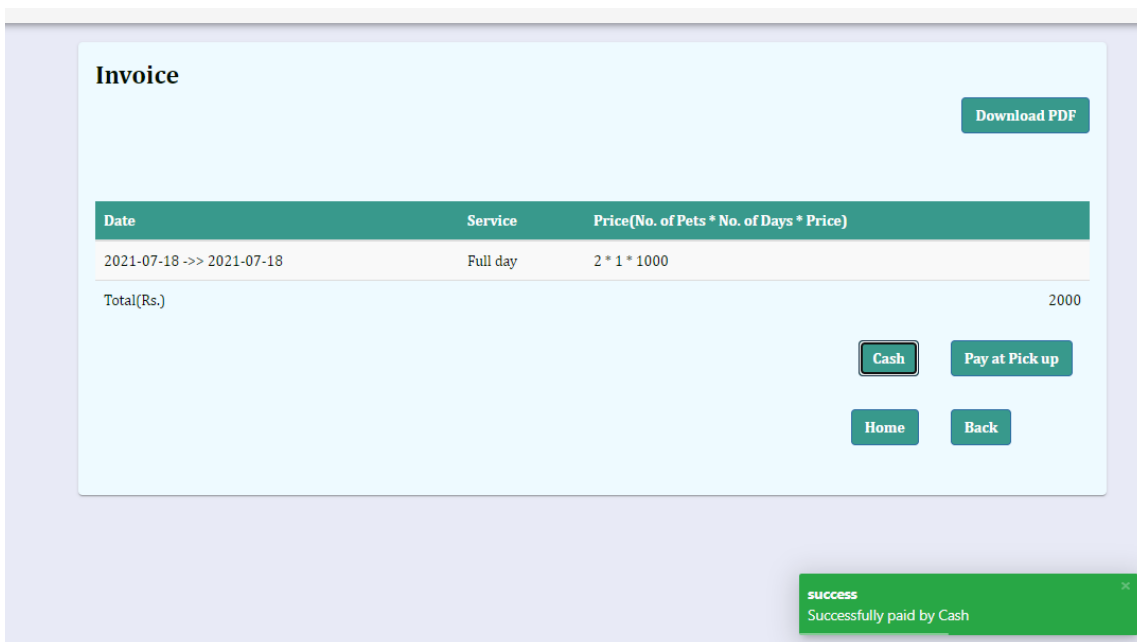


Figure 4.6: Payment Successful Message

Chapter 5-Testing and Evaluation

5.1 Introduction

Software testing is an important part of software quality assurance since it represents the final analysis of a software product's specification, design, and code creation. Verification and Validation are fundamentally combined in the testing technique. Validation is a dynamic mechanism for testing and validating whether the software product fits the customer's exact demands or not, whereas Verification is a process of checking documentation, design, code, and program to see if the software was constructed according to the requirements or not (Guru99.com. 2021). The primary purpose of user evaluation is to determine whether the customer's needs are being satisfied to an acceptable standard.

5.2 Testing Procedure

This software has been tested by following the manual testing as well as the test automation.

5.2.1 Manual Testing

Following are testing types used for manual testing (Applause. 2021).

- Unit testing: Each module of the system is tested individually in order to confirm the functionality of particular module.
- Integration testing : Several modules of the system combined and tested
- System testing: System as a whole is tested
- Retesting : After fixing identified bugs particular feature will be tested again to pass the failed test cases
- Regression testing: After fixed bugs, check whether existing functionalities are working as expected without introducing new bugs

5.2.2 Test Automation

Appointment creation module will be automated as it's the core feature of this system. The framework of automation will be developed using Selenium with Java using Page Object Model.

5.3 Test Plan and Test Cases

Test objectives, timetable and logistics, test methodologies, and, most importantly, test cases are included in test plan. Test cases were written in accordance with the test plan. This includes a description of the test, the expected outcome, and the current progress of each test scenario (SOFTWARE TESTING Fundamentals. 2021). To make the testing process less complicated, test cases were created separately for each module. Some test cases are listed in the tables below.

Test Case ID		001	
Test Module		Login	
Test Case		Verify Login functionality	
No	Test Description	Expected Result	Status
1	Verify valid login	User should be able to login to the system with valid username and password	Pass
2	Verify invalid login	User should not be able to login to the system with invalid username and/or password	Pass

Table 5.1: Test Case-Verify Login functionality

Test Case ID 002			
Test Module	Profile		
Test Case	Create Profile		
Prerequisites	User should be logged into the system as a customer		
No	Test Description	Expected Result	Status
1	Verify successful creation of customer profile	Customer Profile should be created	Pass
2	Verify add pet functionality	Added pets should be displayed in View tab	Pass
3	Validation of fields	Error message should be displayed if mandatory fields are not filled. Error message should be displayed if validation is failed for NIC, Email and Phone number	Pass

Table 5.2: Test Case-Create Profile

Test Case ID 003			
Test Module	Profile		
Test Case	Edit Profile		
Prerequisites	Successfully created Customer Profile		
No	Test Description	Expected Result	Status
1	Verify edit functionality of customer profile	Customer should be able to edit customer profile details	Pass
2	Verify edit pet functionality	Customer should be able to edit pet details	Pass

3	Verify add/edit medication details	Customer should be able to add/edit medication details	Pass
4	Verify add/edit dietary details	Customer should be able to add/edit dietary details	Pass
5	Verify add/edit vaccination details	Customer should be able to add/edit vaccination details	Pass

Table 5.3: Test Case- Edit Profile

Test Case ID	004		
Test Module	Appointments		
Test Case	Create Appointment		
Prerequisites	Successfully created Customer Profile		
No	Test Description	Expected Result	Status
1	Verify create appointment functionality	Customer should be able to create appointment successfully	Pass
2	Validate whether check out date/time>check in date/time	Check out date should be a later date/time than check in date	Pass

Table 5.4: Test Case- Create Appointment

Test Case ID 005			
Test Module	Portal		
Test Case	Checked In /Checked Out Pet		
Prerequisites	Successfully created an appointment		
No	Test Description	Expected Result	Status
1	Verify Checked In functionality	Operator should be able to checked in pet	Pass
2	Verify Checked Out functionality	Operator should be able to checked out pet	Pass

Table 5.5: Test Case- Checked In /Checked Out Pet

Test Case ID 006			
Test Module	Products		
Test Case	Retail		
Prerequisites	User should be logged into the system as a customer		
No	Test Description	Expected Result	Status
1	Verify Add to Cart functionality	Customer should be able to add to cart (One retail)	Pass
2	Verify Add to Cart and Search functionality	Customer should be able to add to cart and Search(Many retails)	Pass

Table 5.6: Test Case- Retail

Test Case ID 007			
Test Module	Products		
Test Case	Suppliers		
Prerequisites	User should be logged into the system as a supplier		
No	Test Description	Expected Result	Status
1	Verify Add Supplier details	Supplier should be able to add details	Pass

Table 5.7: Test Case- Suppliers

Test Case ID 008			
Test Module	Pet Live Board		
Test Case	Pet Live Board-Check in Pet for current date		
Prerequisites	User should be logged into the system as an operator/admin		
No	Test Description	Expected Result	Status
1	Verify Pet box with appointment details for pets who are checked in for the current date	Operator/admin should be able to view details of pets who are checked in for the current date	Pass

Table 5.8: Test Case- Pet Live Board-Check in Pet for current date

Test Case ID 009			
Test Module	Employees		
Test Case	Add/Edit Employee		
Prerequisites	User should be logged into the system as an operator		
No	Test Description	Expected Result	Status
1	Verify add employee details functionality	Operator should be able to add employee details	Pass
	Verify edit employee details functionality	Operator should be able to edit employee details	Pass

Table 5.9: Test Case- Add/Edit Employee

Test Case ID 010			
Test Module	Employees		
Test Case	Employee schedule		
Prerequisites	User should be logged into the system as an operator		
No	Test Description	Expected Result	Status
1	Verify add employee schedule for each employee functionality	Operator should be able to add employee schedules	Pass

Table 5.10: Test Case- Employee schedule

Test Case ID 011			
Test Module	Veterinary		
Test Case	Add/Edit Veterinarian		
Prerequisites	User should be logged into the system as an operator		
No	Test Description	Expected Result	Status
1	Verify add Veterinarian functionality	Operator should be able to add Veterinarian	Pass
2	Verify edit Veterinarian functionality	Operator should be able to edit Veterinarian	Pass

Table 5.11: Test Case- Add/Edit Veterinarian

Test Case ID 012			
Test Module	Service Dogs		
Test Case	Add/Edit/Delete Service Dogs		
Prerequisites	User should be logged into the system as an operator		
No	Test Description	Expected Result	Status
1	Verify add service dogs functionality	Operator should be able to add service dogs	Pass
2	Verify edit service dogs functionality	Operator should be able to edit service dogs	Pass
3	Verify delete service dogs functionality	Operator should be able to delete service dogs	Pass

Table 5.12: Test Case- Add/Edit/Delete Service Dogs

Test Case ID 013			
Test Module	Reports		
Test Case	Feeding Report		
Prerequisites	User should be logged into the system as admin		
No	Test Description	Expected Result	Status
1	Verify report is displayed for selected date	Admin should be able to view feeding details of pets	Pass
2	Verify download functionality	Admin should be able to download report	Pass

Table 5.13: Test Case- Feeding Report

Test Case ID 014			
Test Module	Reports		
Test Case	Medication Report		
Prerequisites	User should be logged into the system as admin		
No	Test Description	Expected Result	Status
1	Verify report is displayed for selected date	Admin should be able to view medication details of pets	Pass
2	Verify download functionality	Admin should be able to download report	Pass

Table 5.14: Test Case- Medication Report

Test Case ID 015			
Test Module	Reports		
Test Case	Check In Report		
Prerequisites	User should be logged into the system as admin		
No	Test Description	Expected Result	Status
1	Verify report is displayed for selected date range	Admin should be able to view check in details of pets	Pass
2	Verify download functionality	Admin should be able to download report	Pass

Table 5.15: Test Case- Check In Report

Test Case ID 016			
Test Module	Reports		
Test Case	Check Out Report		
Prerequisites	User should be logged into the system as admin		
No	Test Description	Expected Result	Status
1	Verify report is displayed for selected date range	Admin should be able to view check out details of pets	Pass
2	Verify download functionality	Admin should be able to download report	Pass

Table 5.16: Test Case- Check Out Report

Test Case ID 017			
Test Module	Reports		
Test Case	Revenue Report		
Prerequisites	User should be logged into the system as admin		
No	Test Description	Expected Result	Status
1	Verify report is displayed for selected date range	Admin should be able to view revenue details	Pass
2	Verify download functionality	Admin should be able to download report	Pass

Table 5.17: Test Case- Revenue Report

Test Case ID 018			
Test Module	Reports		
Test Case	List of Service Dogs		
Prerequisites	User should be logged into the system as admin		
No	Test Description	Expected Result	Status
1	Verify service dog details are displayed	Admin should be able to service dog details	Pass
2	Verify download functionality	Admin should be able to download report	Pass

Table 5.18: Test Case- List of Service Dogs

5.3 Sample Test Results

Following Figure 5.1 to 5.16 show sample test results.

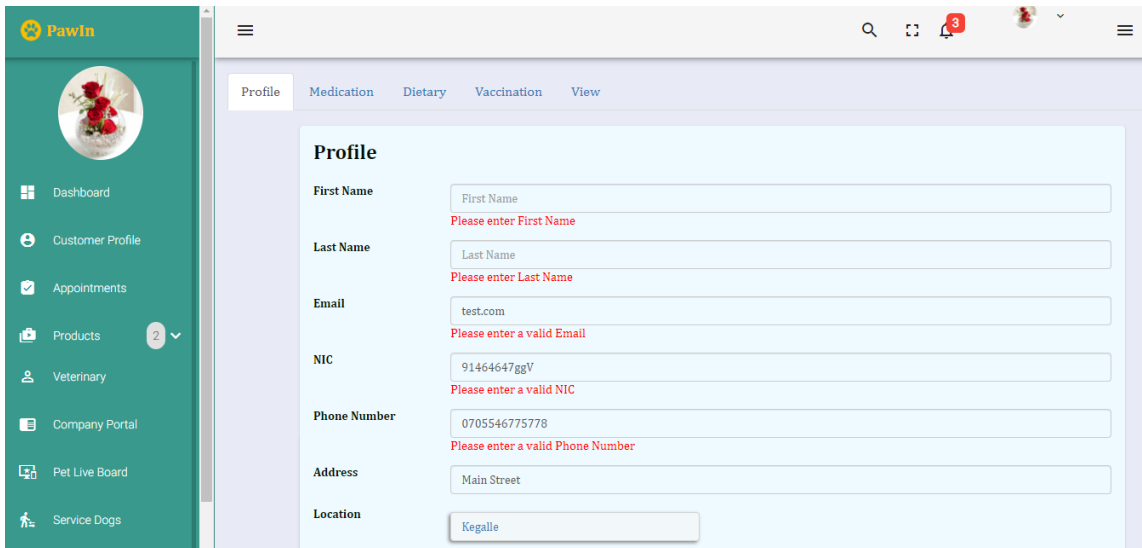


Figure 5.1: Validation of fields

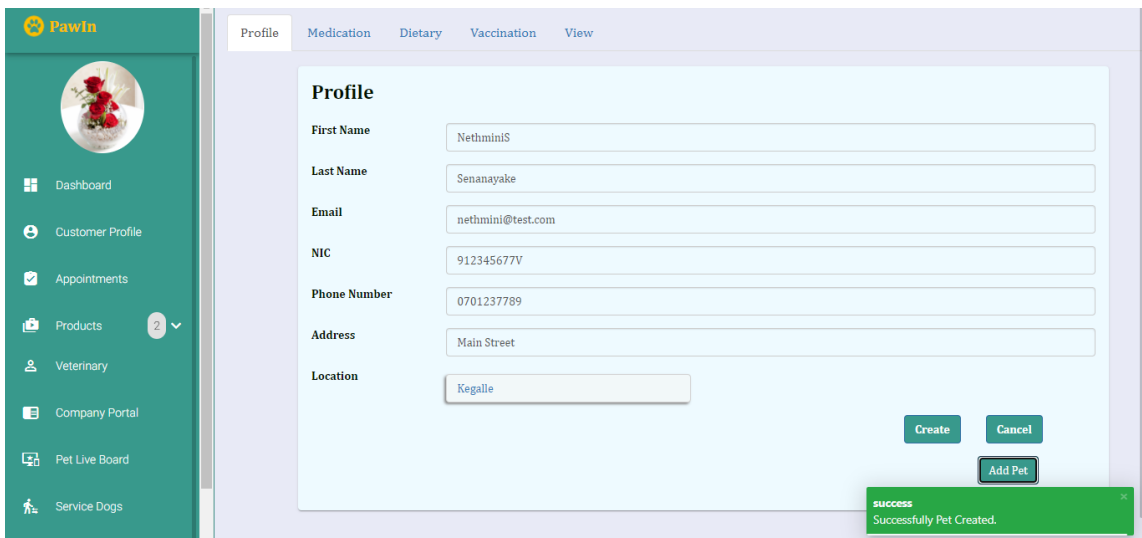


Figure 5.2: Create Pet

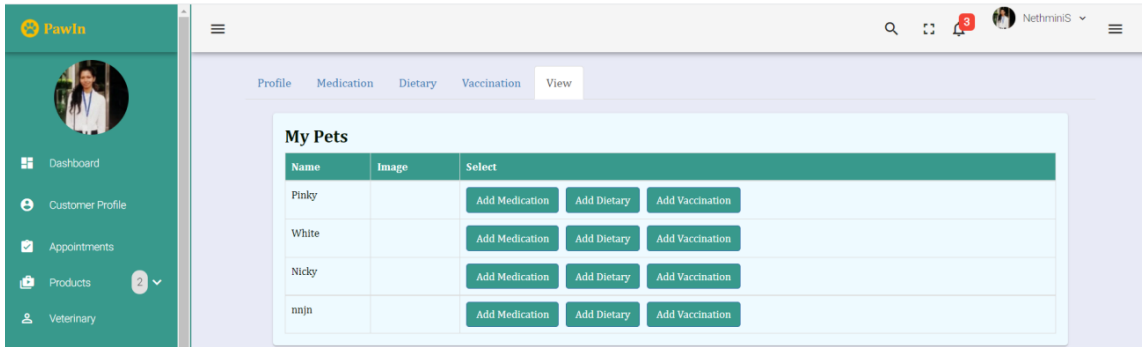


Figure 5.3: View Pet

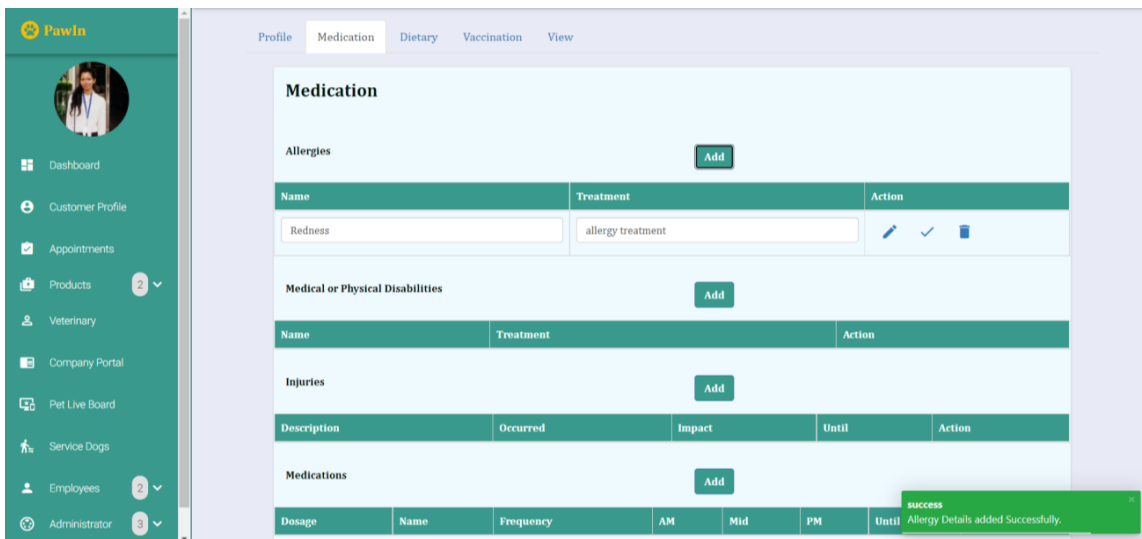


Figure 5.4: Add/view allergy details to particular Pet

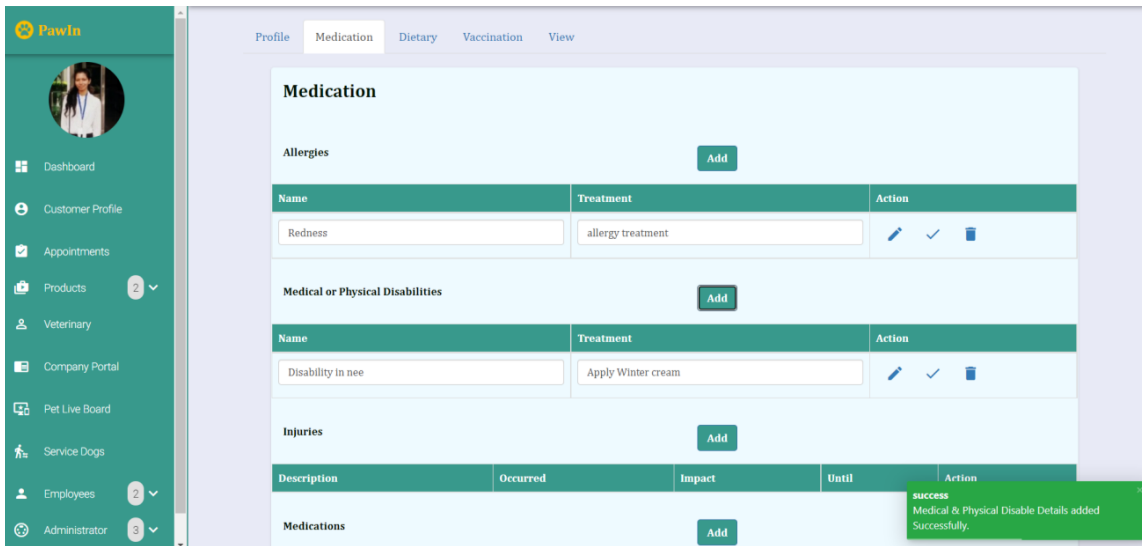


Figure 5.5: Add/view medical and physical disabilities' details to particular Pet

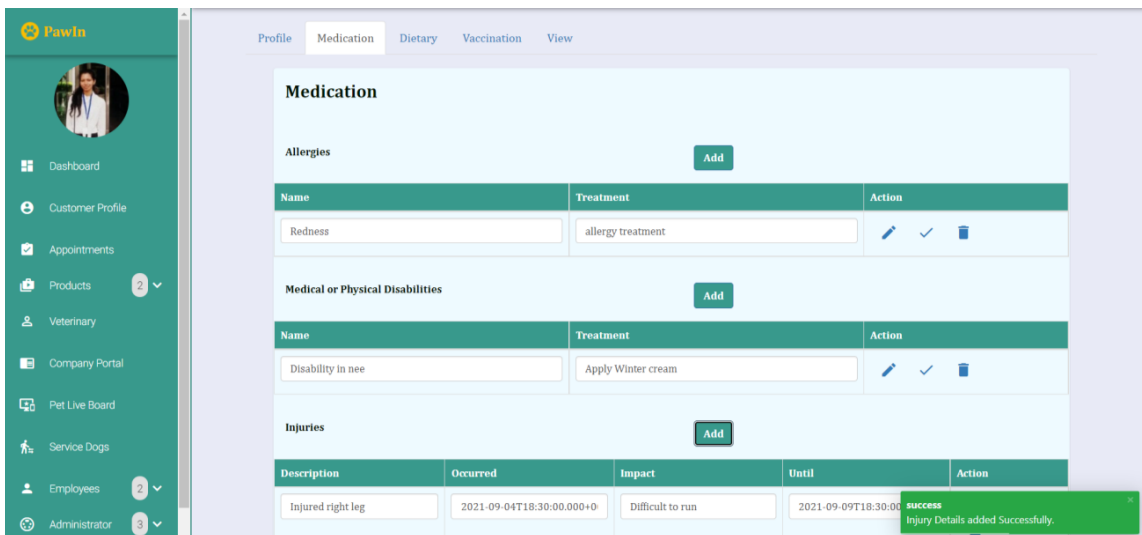


Figure 5.6: Add/view injury details to particular Pet

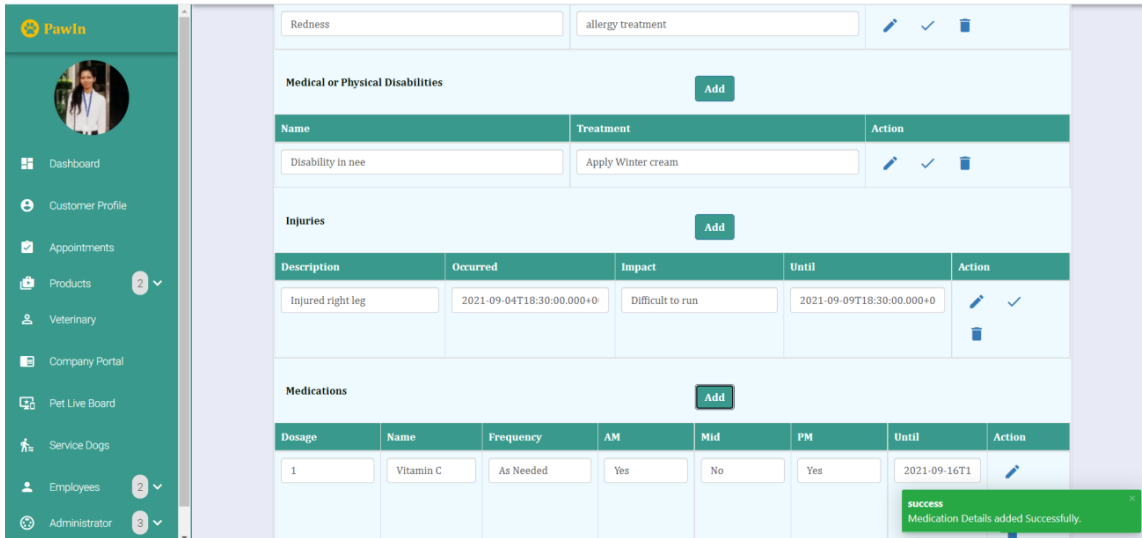


Figure 5.7: Add/view medication details to particular Pet

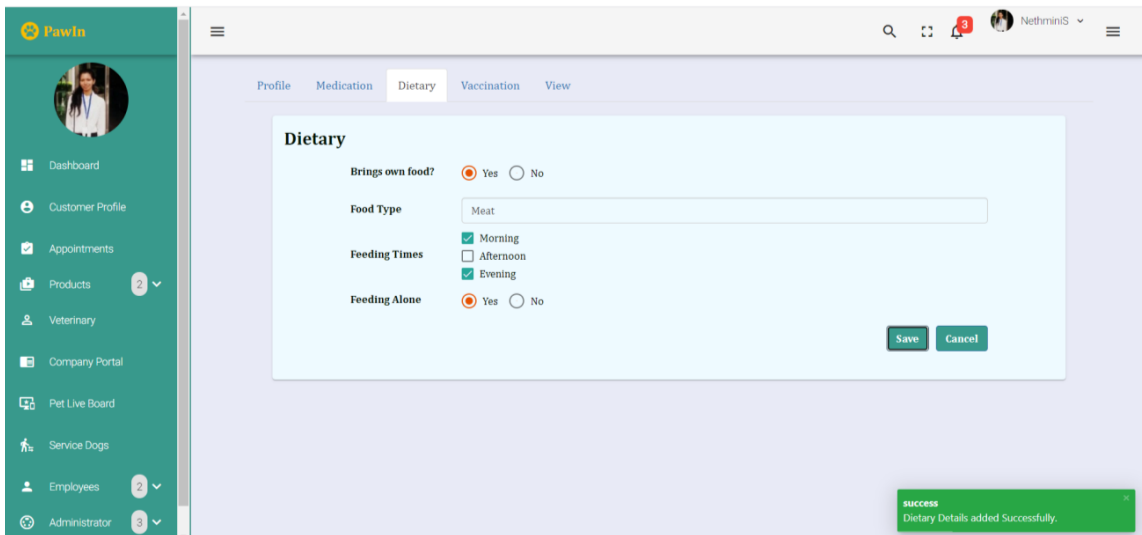


Figure 5.8: Add dietary details to particular Pet

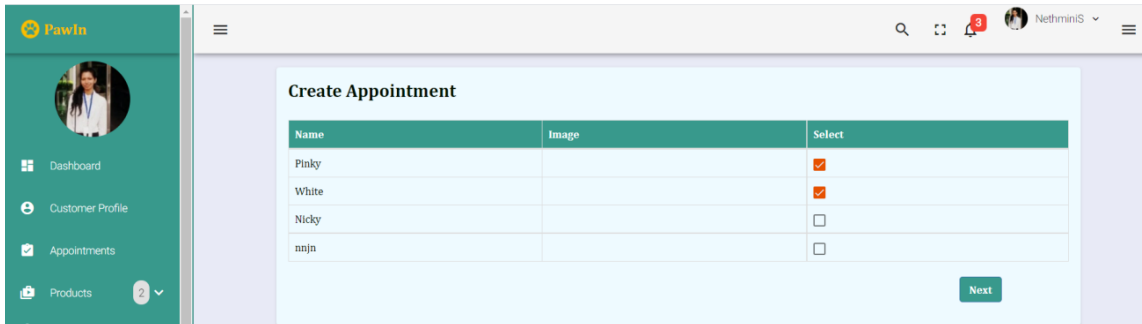


Figure 5.9: Create Appointment-Select Pets

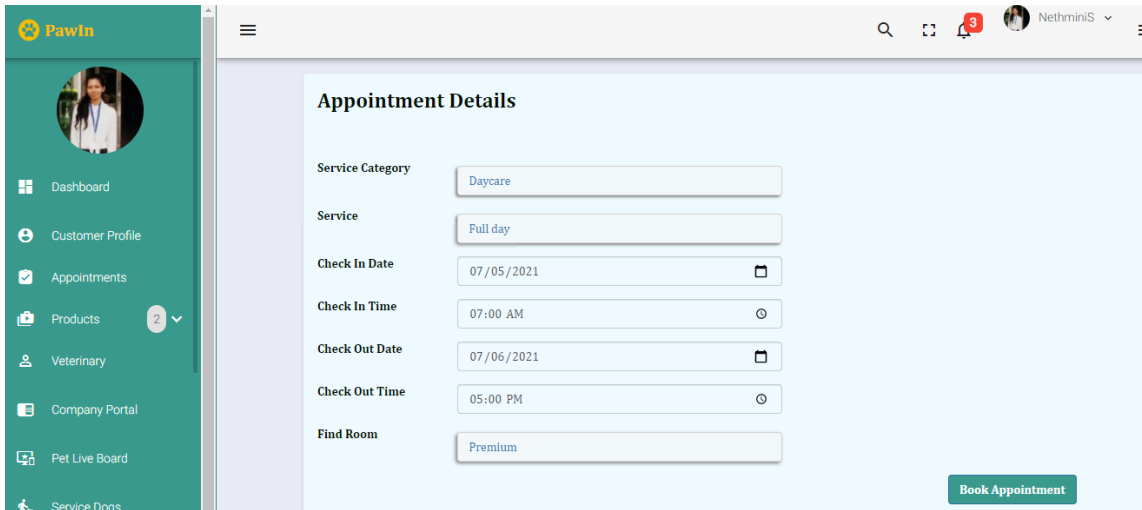


Figure 5.10: Create Appointment-Enter Appointment details (Resource based)

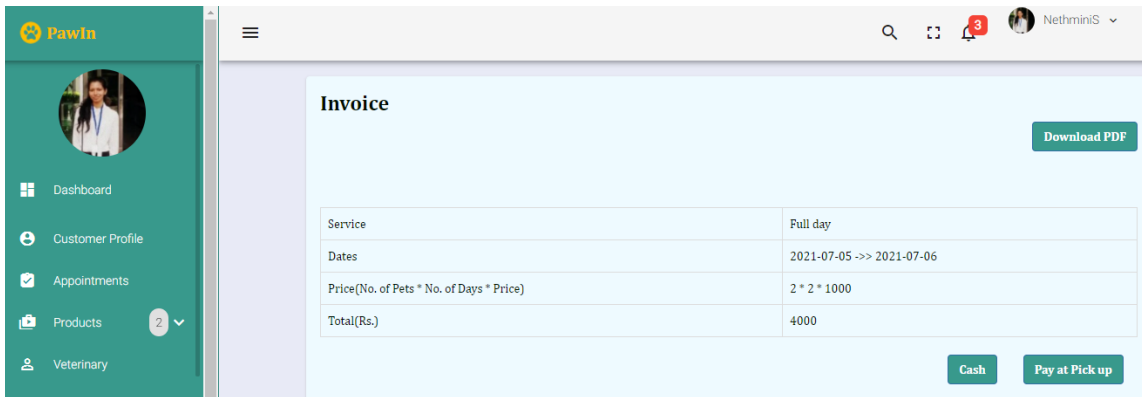


Figure 5.11: Create Appointment-Invoice

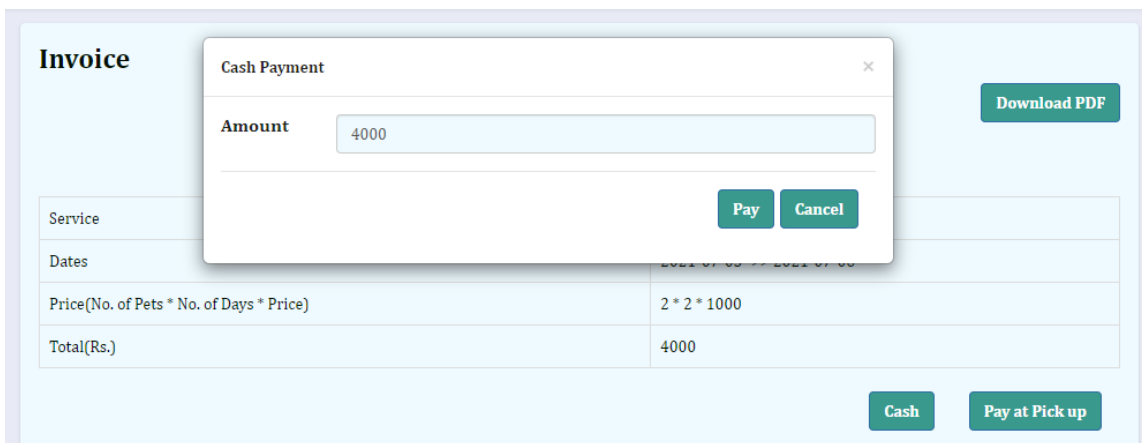


Figure 5.12: Create Appointment-Pay by Cash

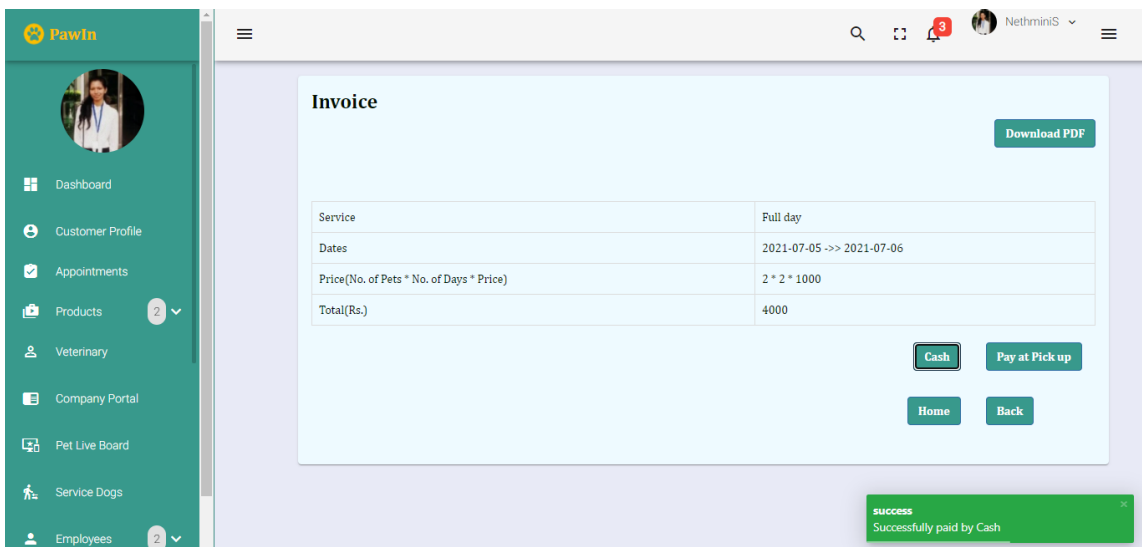


Figure 5.13: Create Appointment successfully

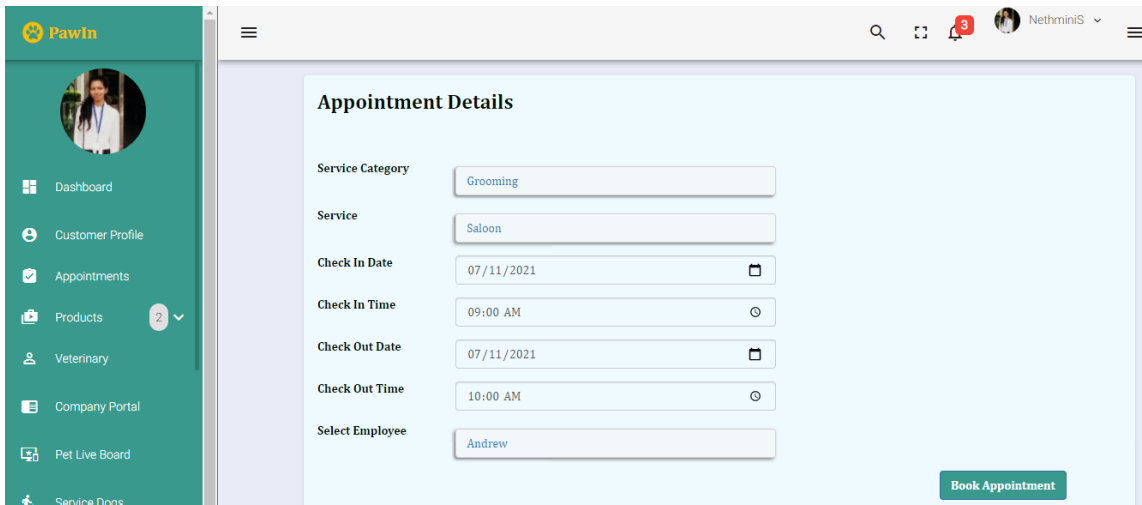


Figure 5.14: Create Appointment-Enter Appointment details (Employee based)

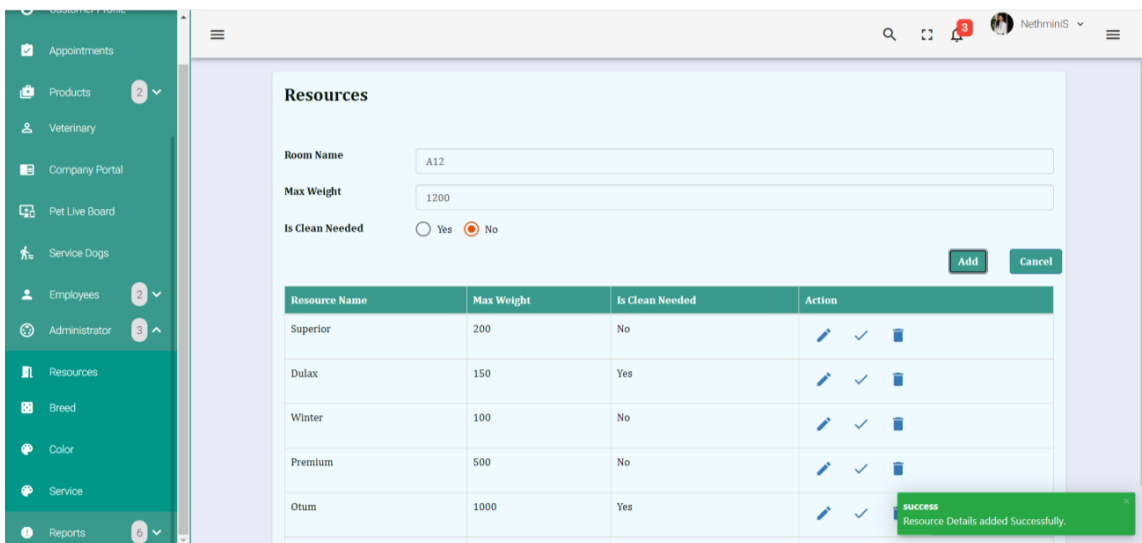


Figure 5.15: Add Resource

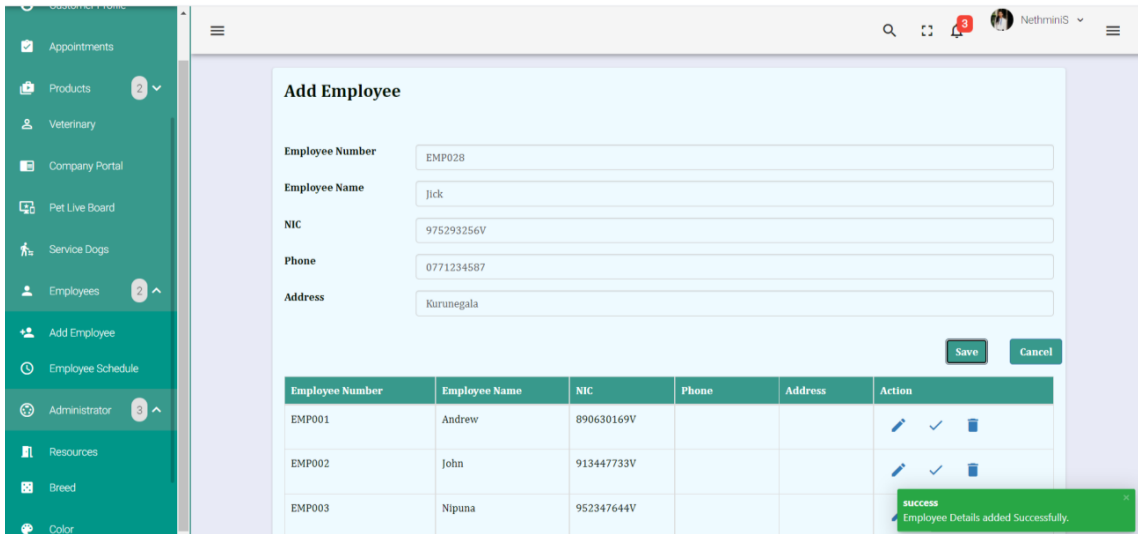
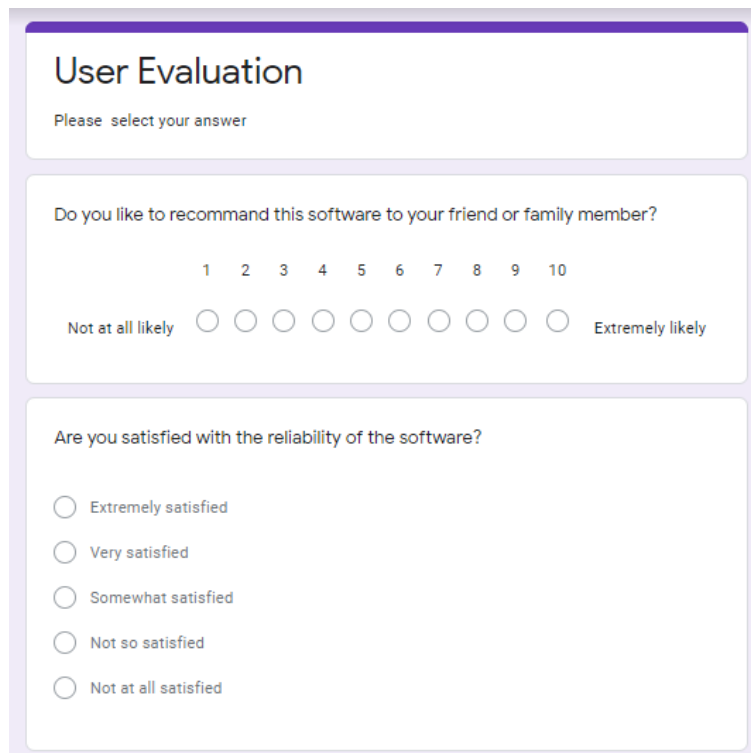


Figure 5.16: Add Employee

5.4 User Evaluation

This system has been uploaded to a common platform as a product with limited features. Then required pet institutions can use those limited features. User evaluation was conducted via a Google questionnaire by sending it to the pet institutions which have selected the product for use.

Following Figure 5.17 to 5.21 demonstrate the questionnaire.



The image shows a digital questionnaire titled "User Evaluation". It contains two questions. The first question is "Do you like to recommend this software to your friend or family member?" and is answered using a 10-point Likert scale from "Not at all likely" to "Extremely likely". The second question is "Are you satisfied with the reliability of the software?" and is answered using a five-point Likert scale from "Extremely satisfied" to "Not at all satisfied".

User Evaluation
Please select your answer

Do you like to recommend this software to your friend or family member?

1 2 3 4 5 6 7 8 9 10

Not at all likely Extremly likely

Are you satisfied with the reliability of the software?

Extremely satisfied
 Very satisfied
 Somewhat satisfied
 Not so satisfied
 Not at all satisfied

Figure 5.17: User Evaluation Form (Q1-Q2)

How satisfied are you with the security of the software?

- Extremely satisfied
- Very satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

How satisfied are you with the software's ease of use ?

- Extremely satisfied
- Very satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

Figure 5.18: User Evaluation Form (Q3-Q4)

Are you satisfied with the features of the software?

- Extremely satisfied
- Very satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

Are you satisfied with the look and feel of the software?

- Extremely satisfied
- Very satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

Figure 5.19: User Evaluation Form (Q5-Q6)

How satisfied with the account set up experience of the software?

- Extremely satisfied
- Very satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

How satisfied are you with the value for the money of the software?

- Extremely satisfied
- Very satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

Figure 5.20: User Evaluation Form (Q7-Q8)

Are you satisfied with the performance of the software?

- Extremely satisfied
- Very satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

Do you have any ideas of how to improve the software?

.....

Submit

Figure 5.21: User Evaluation Form (Q9-Q10)

:

Chapter 6- Conclusion

6.1 Introduction

This chapter indicates the problems encountered during the process of software development, the way overcame the mentioned problems, lesson learnt and finally the future enhancements of the developed system.

6.2 Problems Encountered

Most of the similar domain knowledge was gathered via systems or websites available in the web. Therefore, the researcher had to get in depth knowledge by studying several similar systems in global and local context in order to be more familiar with the domain. After learnt the expected domain knowledge, the researcher decided to make a common product based on the features gathered from different sources. Service Oriented Architecture (SOA) was selected as the architecture initially but it has been changed to Micro services architecture due to the flexibility after analyzed the drawbacks of SOA carefully. When developing the system, the initial backend and front end created had been changed sometimes because of the complexity behind the application. The researcher has overcome this situation by dividing the components into smaller sections and drawing a scratch on a paper.

6.3 Lessons Learnt

The researcher attained a good opportunity to grab the technical knowledge and hands on experience from framework level. The process of developing the gathered requirements by minimizing the complexity of the application was learned. This project provided an opportunity to apply the most important theories and technologies learned during the MIT degree program practically. It also inspired me to research various new and updated technologies like angular, spring boot, and REST-Assured in order to improve system speed. In addition, considerable attention was paid to learning the Dependency Injection design pattern.

6.4 Future enhancements

- Introducing an android application

Through an android application pet owners who are willing to look after their pets for a specific time period can create their profiles without using a PC or a laptop. With the busy schedules of the pet owners it will be a better motivation for them. Suppliers will have the ability to supply retails via the android application effortlessly. Pet intuitions will also manage the process in any location with the use of an android application.

- Extend the application to SMS Services

As a future enhancement for the proposed solution this can be extended to the process of SMS services. By this, vaccination reminder emails and birthday emails can also be sent as texts. The process of SMS can be completed without using the internet service.

- Introducing Payment Gateway

Online payment gateway can be introduced for credit/debit card payments as a payment option for create appointment and add retail functions. This would be a good opportunity for pet owners and suppliers to make their lives easy.

References

[1]Angular.io. 2021. Angular. [online] Available at: <<https://angular.io/>> [Accessed 19 May 2021].

[2]Applause. 2021. *15 Functional Testing Types with Examples*. [online] Available at: <<https://www.applause.com/blog/functional-testing-types-examples>> [Accessed 18 June 2021].

[3]Bestcare.lk. 2021. *Best Care Animal Hospital – Veterinary services Sri Lanka | Pet Boarding Services | house calls veterinary service*. [online] Available at: <<https://bestcare.lk/>> [Accessed 19 May 2021]

[4]BookingKoala. 2021. *Sign Up For Free!*. [online] Available at: <<https://www.bookingkoala.com/>> [Accessed 19 May 2021].

[5]Barkstreet.lk. 2021. *Home - Bark Street Sri Lanka*. [online] Available at: <<http://www.barkstreet.lk/>> [Accessed 19 May 2021].

[6]Galipeau, N., Galipeau, N. and Galipeau, N., 2021. *ProPet Software | Boarding Kennel, Daycare, Training, and Grooming Software*. [online] Propetware.com. Available at: <<https://www.propetware.com/>>[Accessed 19 May 2021].

[7]Guru99.com. 2021. *Difference Between Verification and Validation with Example*. [online] Available at: <<https://www.guru99.com/verification-v-s-validation-in-a-software-testing.html>> [Accessed 18 June 2021].

[8]Guru99.com. 2021. *How to Write Test Cases: Sample Template with Examples*. [online] Available at: <<https://www.guru99.com/test-case.html>> [Accessed 18 June 2021].

[9] microservices.io. 2021. *Microservices Pattern: Microservice Architecture pattern*. [online] Available at: <<https://microservices.io/patterns/microservices.html>> [Accessed 19 May 2021].

- [10]Medium. 2021. *The What, Why, and How of a Microservices Architecture*. [online] Available at: <<https://medium.com/hashmapinc/the-what-why-and-how-of-a-microservices-architecture-4179579423a9>> [Accessed 19 May 2021].
- [11]Petcare, P., 2021. *Professional Software for Pet Sitters - Precise Petcare*. [online] Precisepetcare.com.
Available at: <<https://www.precisepetcare.com/>> [Accessed 19 May 2021].
- [12]Petexec.net. 2021. *PetExec | The #1 Tool for Pet Care Businesses*. [online] Available at:<<https://www.petexec.net/>> [Accessed 19 May 2021].
- [13]Petsitclick.com. 2021. [online] Available at: <<https://www.petsitclick.com/>> [Accessed 19 May 2021].
- [14]Revelationpets.com. 2021. *Kennel Software with Free Trial. Best Cloud Based*. [online] Available at: <<https://www.revelationpets.com/>> [Accessed 19 May 2021].
- [15]Smashing Magazine. 2021. *10 Simple and Impressive Design Techniques — Smashing Magazine*. [online] Available at: <<https://www.smashingmagazine.com/2009/04/10-impressive-simple-design-techniques/>> [Accessed 19 May 2021].
- [16]SOFTWARE TESTING Fundamentals. 2021. *Test Plan - SOFTWARE TESTING Fundamentals*. [online] Available at: <<https://softwaretestingfundamentals.com/test-plan/>> [Accessed 18 June 2021].

Appendix A - Design Documentation

Use Case Diagrams of the Proposed System

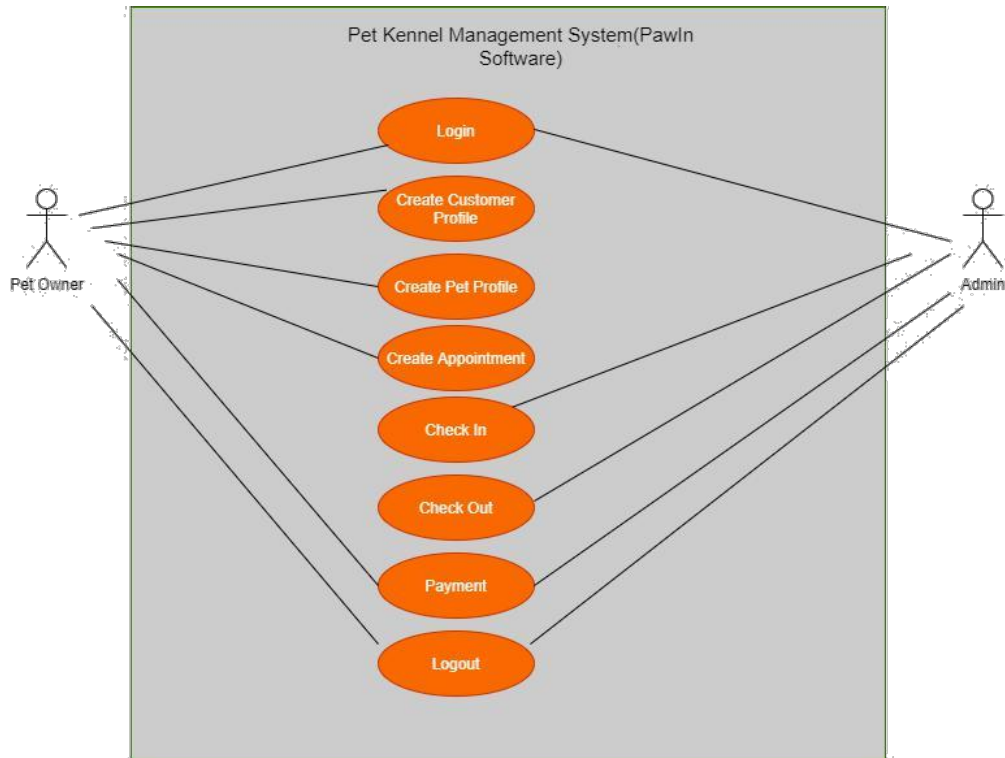


Figure A.1: Use Case Diagram 1 of Proposed Solution

Use case	Appointments
Actors	Pet Owner, Admin
Overview	
Create appointment, checked in and checked out pets	
Pre conditions	
User should have logged in to the system	
Flow of events	
<ol style="list-style-type: none"> 1. Navigate to Customer Profile 2. Create Customer Profile 3. Add Pet details 4. Create appointment 5. Check in pet 	

6. Check out pet
7. Complete payment
8. Logout from the system

Post conditions

Pet Owner completed the appointment process successfully

Table A.1: Use Case 1

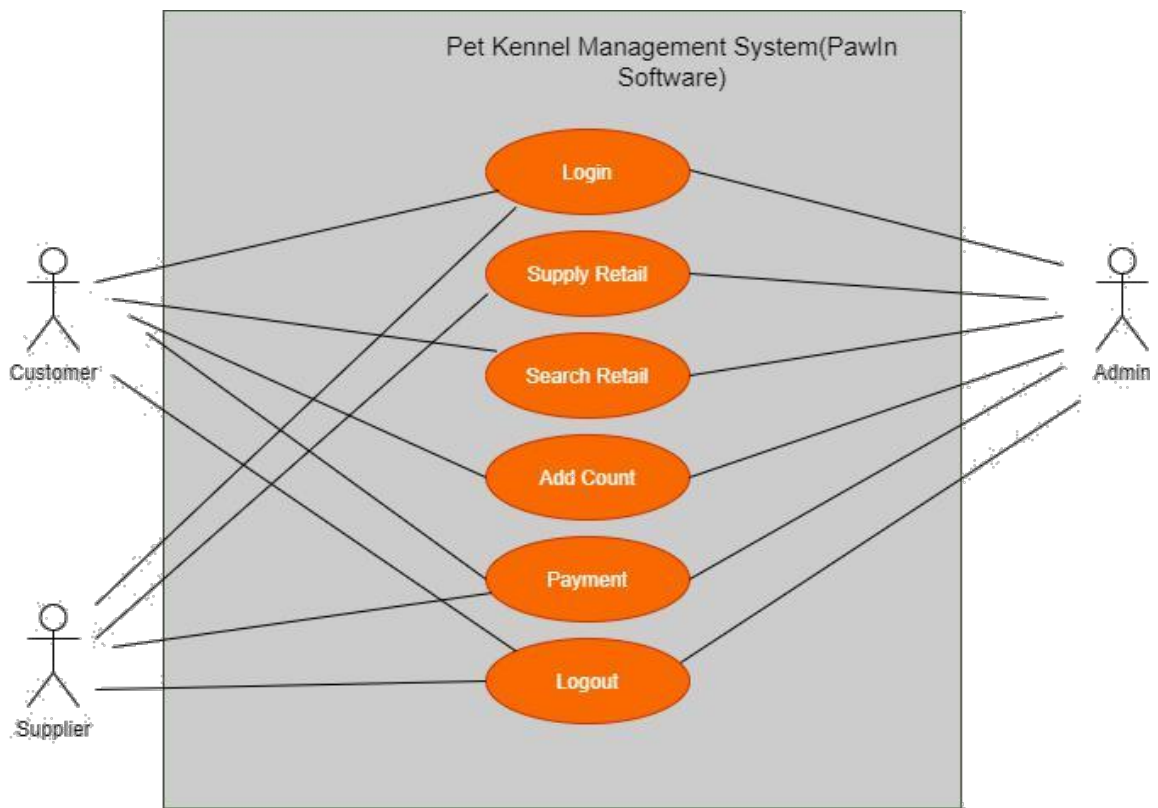


Figure A.2: Use Case Diagram 2 of Proposed Solution

Use case	Retail
Actors	Supplier, Pet Owner, Admin
Overview	
Purchase Retail	
Pre conditions	
User should have logged in to the system	
Flow of events	
<ol style="list-style-type: none"> 1. Supplier register in the system 2. Supplier add retail to the stock 3. Admin approve the retail 4. Pet Owner purchase retail 5. Pet Owner make payment 6. Logout from the system 	
Post conditions	
Purchase retail successfully	

Table A.2: Use Case 2

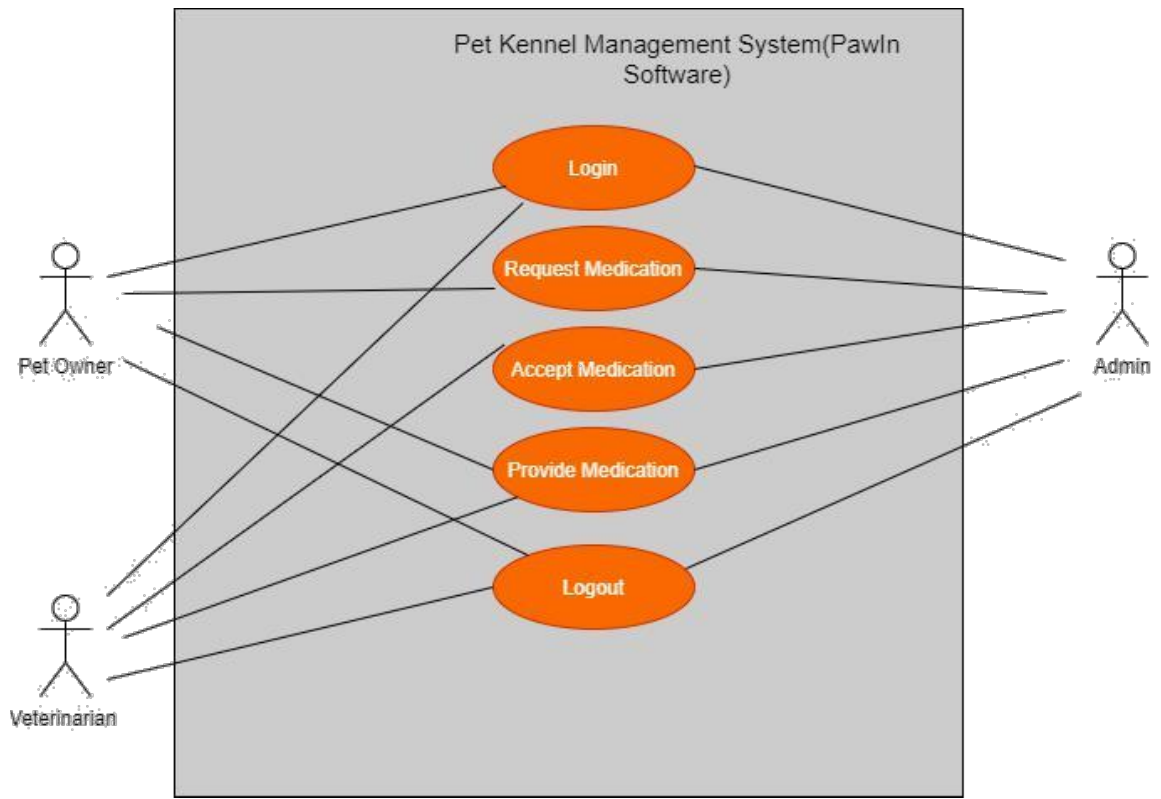


Figure A.3: Use Case Diagram 3 of Proposed Solution

Use case	Veterinary
Actors	Veterinarian, Pet Owner, Admin
Overview	
Maintain medication for pets	
Pre-conditions	
User should have logged in to the system	
Flow of events	
<ol style="list-style-type: none"> 1. Admin register Veterinarian 2. Pet owner request medication when creating customer profile 3. Veterinarian accept medication 4. Veterinarian provide medication 5. Logout from the system 	
Post conditions	
Maintain medication for pets successfully	

Table A.3: Use Case 3

Activity Diagram of the Proposed System

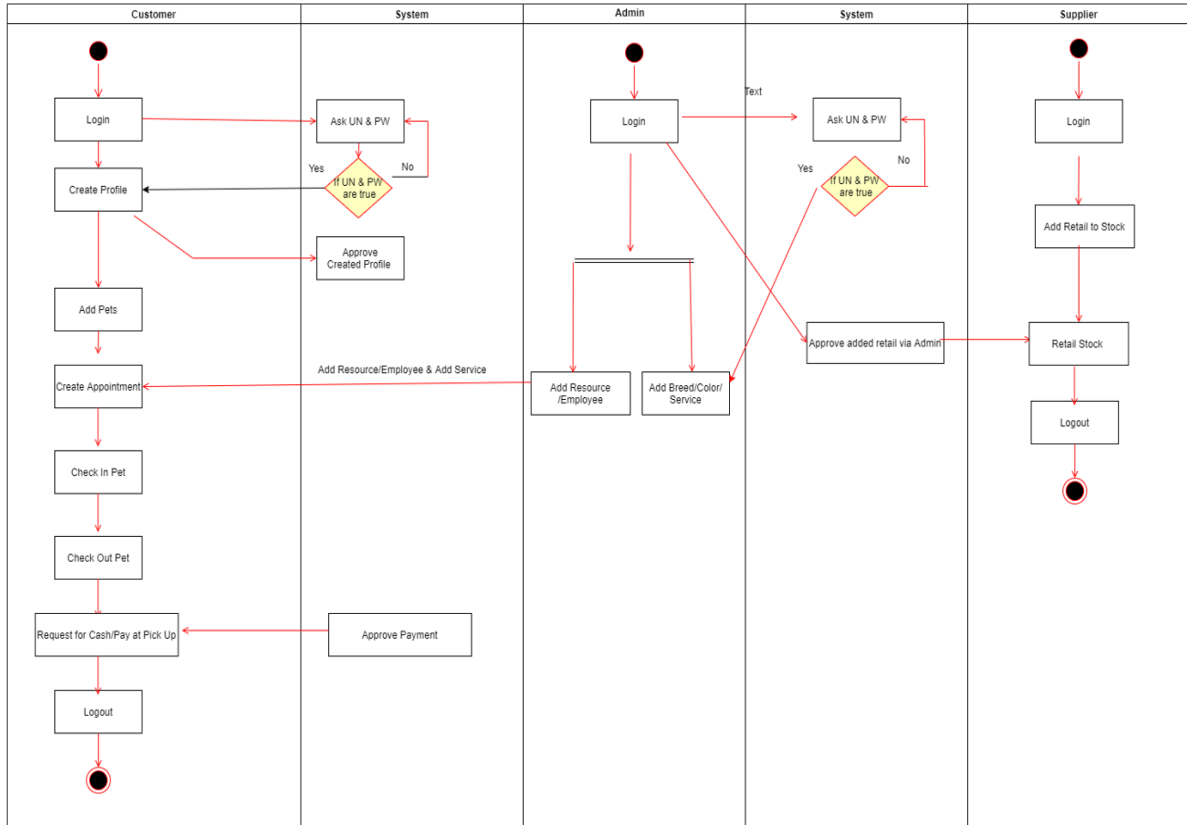


Figure A.4: Activity Diagram of Proposed Solution

Appendix B-Management Reports

The software will generate following reports for particular pet institution.

- Feeding
- Medication
- Check In
- Check Out
- Revenue
- List of Service Dogs

Feeding Reports

This will illustrate the feeding details of pets who are scheduled on selected date as check in date in the appointments. Admin can change the date according to the requirement and get the feeding details of pets who are scheduled to be checked in. This would be helpful to get idea about the feeding times, food types and the own foods are brought by the pet owners for their pets. Since feeding is an essential entity in order to look after pets in the stay this report would be more beneficial to the pet institution. Admin will have the ability to download the report as an excel as a reference.

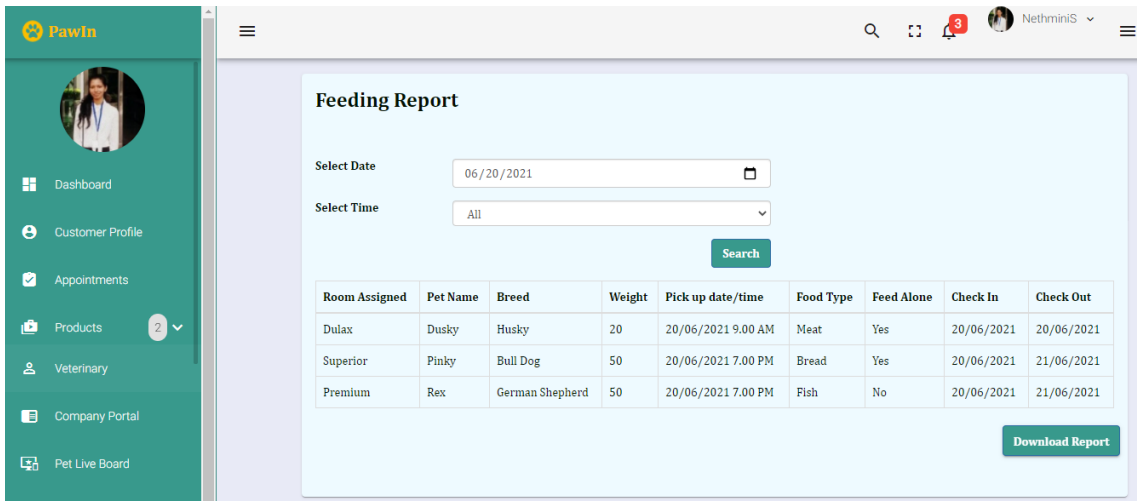


Figure B.1: Feeding Report

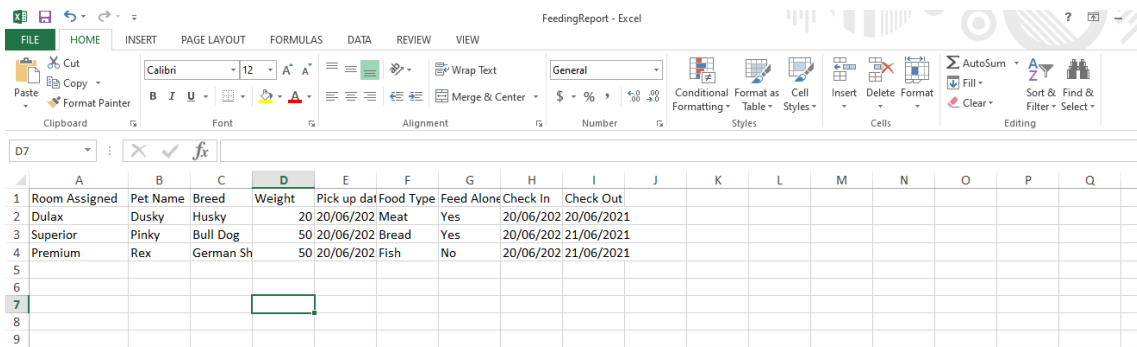


Figure B.2: Feeding Report-Downloaded Report

Medication Reports

This report will generate the any medication that need to be given for particular pet during the stay. The report will supportive to maintain good health of the pets. Admin can select a date and then he will be able to get medication details such as dosage, medication name, frequency, medication times, until which date that need be given for each pet who will be checked in on particular date. Details can be filtered based on medication times such as AM, Mid, PM or All times. Report download facility is also provided for this report.

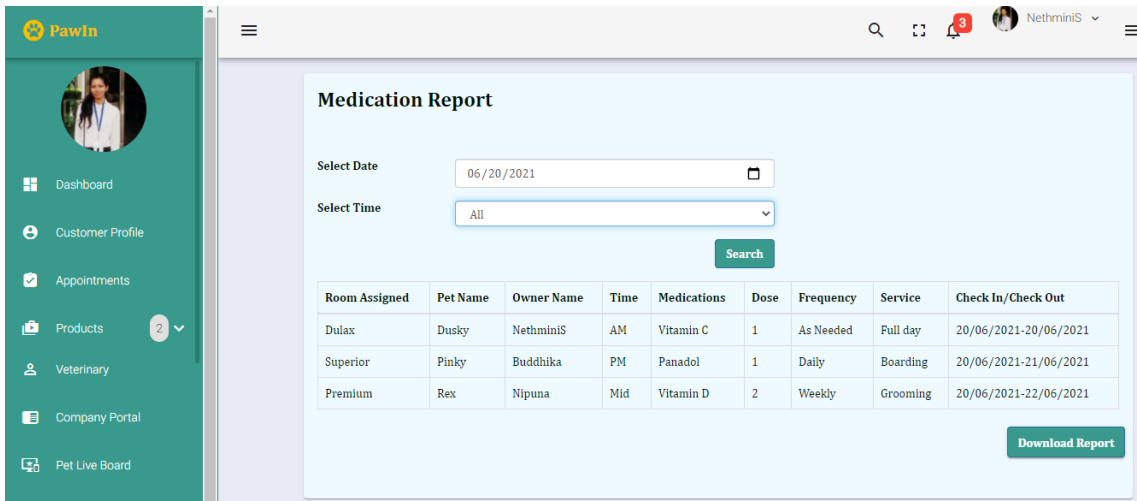


Figure B.3: Medication Report

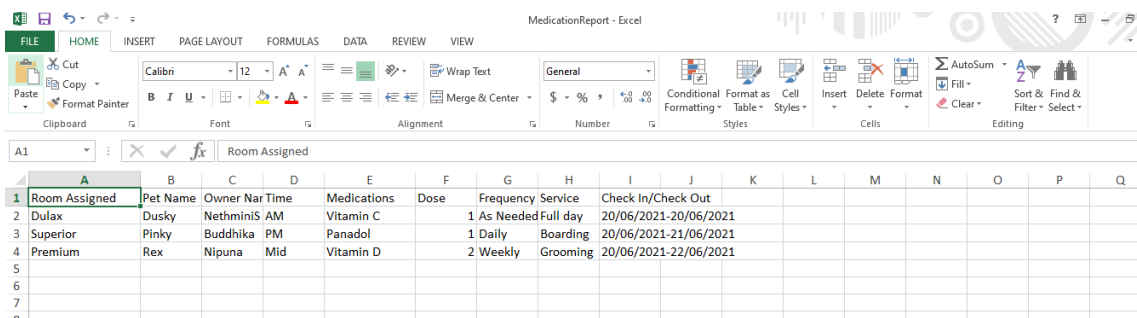


Figure B.4: Medication Report-Downloaded Report

Check In Reports

Pets can be checked in via Portal. When selecting a date admin is able to get details of pets who are already checked in or scheduled to checked in for future dates.

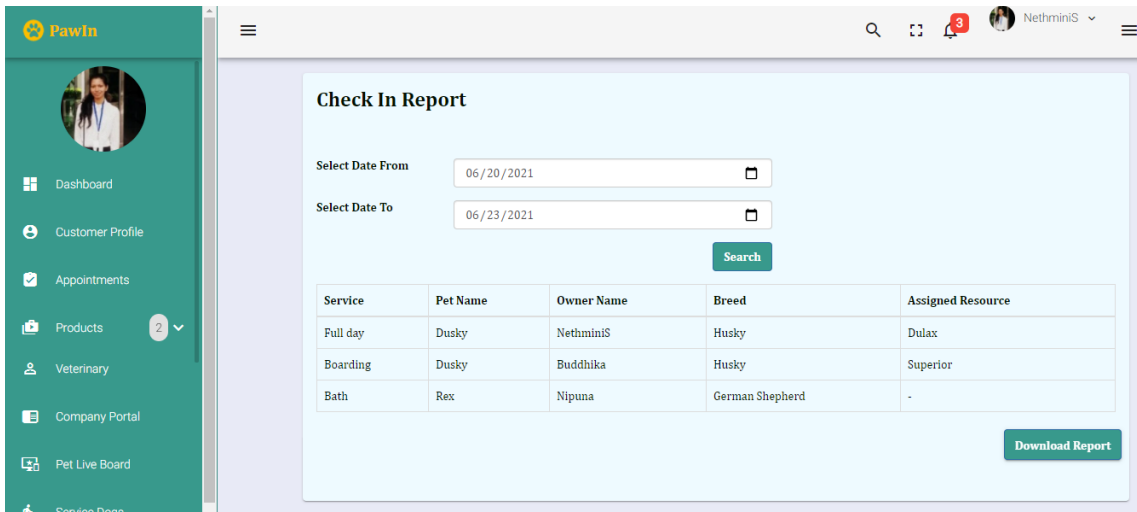


Figure B.5: Check In Report

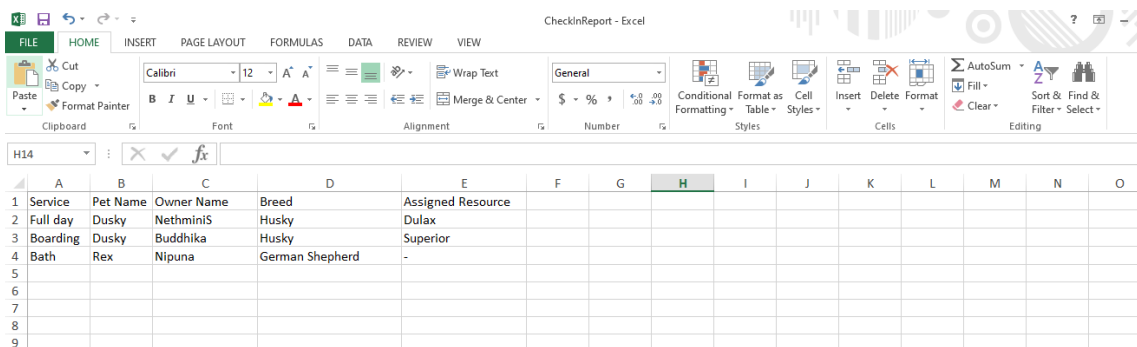


Figure B.6: Check In Report-Downloaded Report

Check Out Reports

Checked in pets can be checked out via Portal. When selecting a date admin is able to get details of pets who are already checked out or scheduled to checked out for future dates.

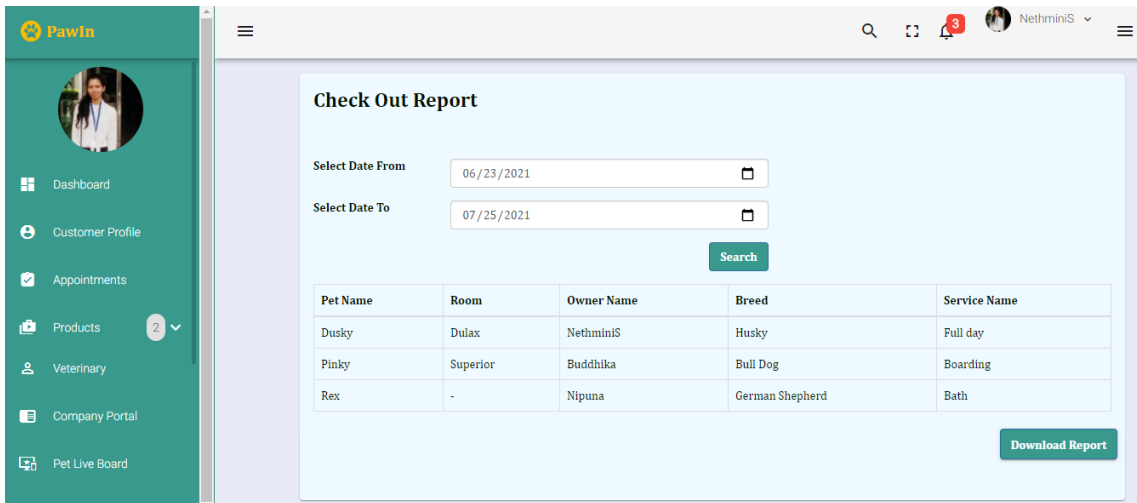


Figure B.7: Check Out Report

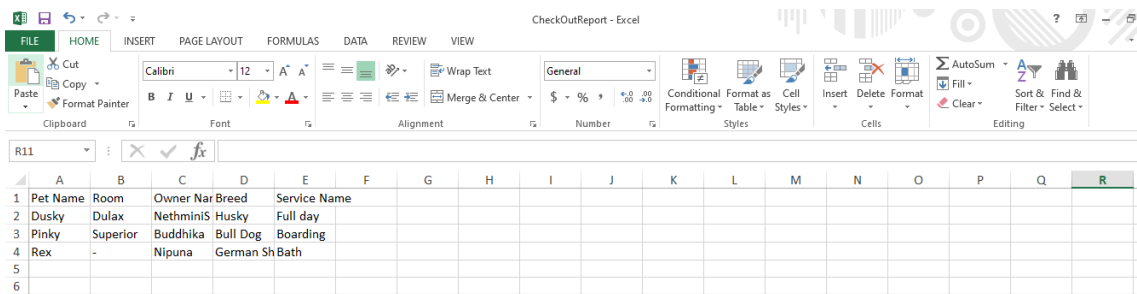


Figure B.8: Check Out Report-Downloaded Report

Revenue Report

Admin can select a date range according to the requirement. Then he will be able to see the income, expense as well as the total revenue during the selected time period. Monthly, quarterly or yearly revenue can be downloaded as an excel for reference.

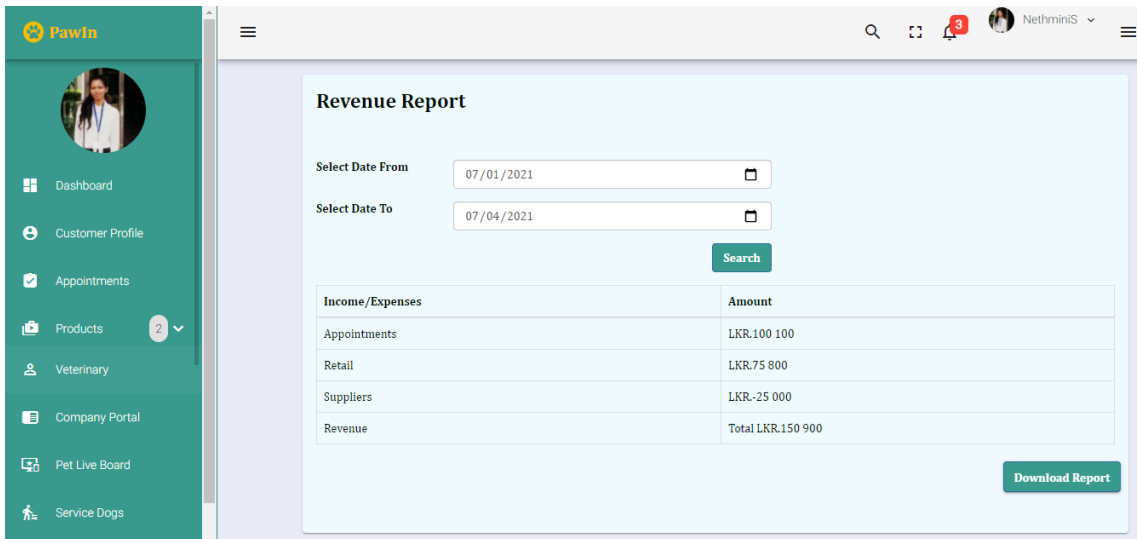


Figure B.9: Revenue Report

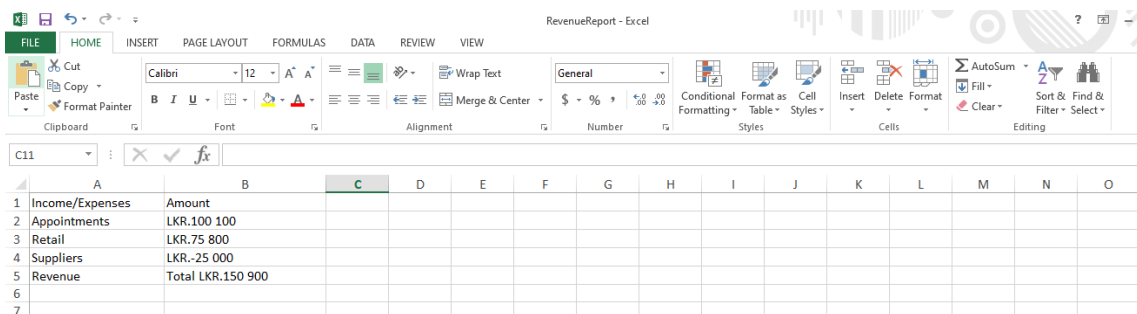


Figure B.10: Revenue Report-Downloaded Report

List of Service Dogs

Dogs who are capable of directing old and blind people can be considered as service dogs. Those are trained by the pet institution and the details of them are also maintained via this system. If customer is requesting a service dog, admin has the ability to provide a list of service dogs who are currently maintaining. This report will facilities that requirement by allowing to download the report and send to the customer.

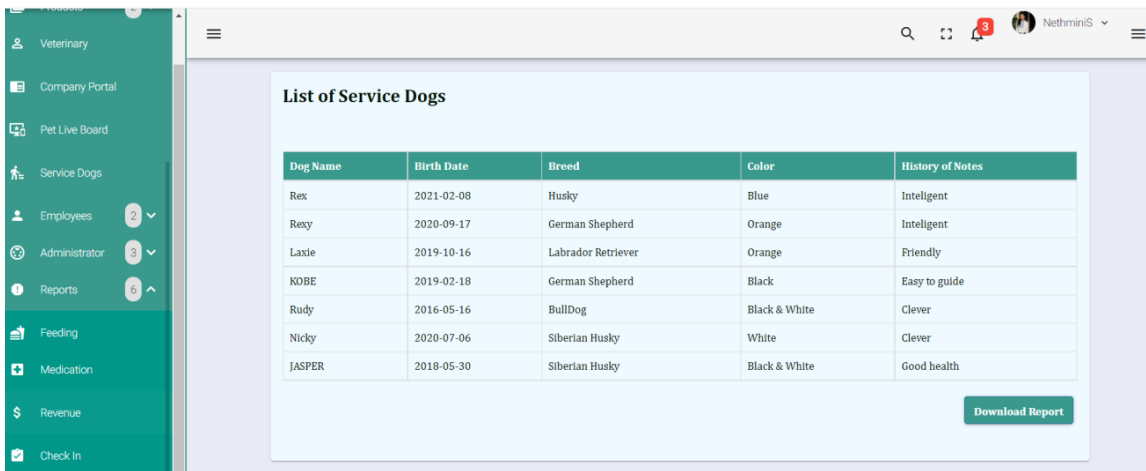


Figure B.11: List of Service Dogs

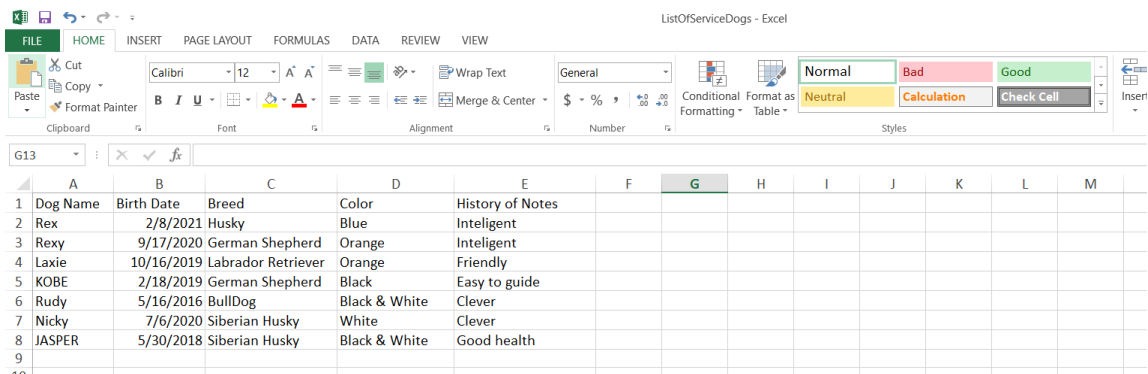


Figure B.12: List of Service Dogs-Downloaded Report

Appendix C – Code Listings

Entity Class-Appointment.java

```
package com.pawin.entity;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import lombok.Data;
import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;

import javax.persistence.*;
import javax.validation.constraints.NotNull;
import java.io.Serializable;
import java.util.Date;

@Entity
@Table(name = "APPOINTMENT")
@EntityListeners(AuditingEntityListener.class)
@JsonIgnoreProperties(value = {"createdAt"},
    allowGetters = true)
@Data
public class Appointment implements Serializable {

    /**
     * 1
     */
    @Id
    @Column(name = "APP_ID")
    @NotNull
    private long appId;

    /**
     * 2
     */
    @Column(name = "APP_CODE", length = 30)
    @NotNull
    private String appCode;

    /**
     * 3
     */
    @Column(name = "SERVICE_CAT_CODE", length = 30)
    private String serviceCatCode;

    /**
     * 4
     */
    @Column(name = "SERVICE_CODE", length = 30)
    private String serviceCode;

    /**
     * 5
     */
    @Column(name = "ROOM_CODE", length = 50)
    private String roomCode;

    /**
     * 6
     */
}
```

```

    */
@Column(nullable = false, updatable = false, name = "CREATED_AT")
@Temporal(TemporalType.TIMESTAMP)
@CreatedDate
private Date createdAt;

/**
 * 7
 */
@Column(name = "CREATED_BY", length = 30)
@NotNull
private String createdBy;

/**
 * 8
 */
@Column(name = "UPDATED_AT")
private Date updatedAt;

/**
 * 9
 */
@Column(name = "UPDATED_BY", length = 30)
private String updatedBy;

/**
 * 10
 */
@Column(name = "EMP_CODE", length = 30)
private String empCode;

/**
 * 11
 */
@Column(name = "CHECK_IN_DATE", length = 100)
private Date checkInDate;

/**
 * 12
 */
@Column(name = "CHECK_IN_TIME", length = 15)
private String checkInTime;

/**
 * 13
 */
@Column(name = "CHECK_OUT_DATE", length = 30)
private Date checkOutDate;

/**
 * 14
 */
@Column(name = "TOTAL_PRICE", length = 30)
private String totalPrice;

@Column(name = "PAYMENT_TYPE", length = 30)
private String paymentType;

```



```
}
```

dto Class –AppointmentDto.java

```
package com.pawin.dto;

import lombok.Data;

import java.util.Date;

@Data
public class AppointmentDto {

    private String appId;
    private String appCode;
    private String serviceCatCode;
    private String serviceCode;
    private String roomCode;
    private String empCode;
    private Date checkInDate;
    private String checkInTime;
    private Date checkOutDate;
    private String checkOutTime;
    private Object petCodes;
    private String totalPrice;
    private String paymentType;

}
```

Model Class-Appointment

```
export class Appointment{

    constructor() {
    }

    appCode: any;
    serviceCatCode: any;
    serviceCode: any;
    roomCode: any;
    empCode: any;
    checkInDate: any;
    checkInTime: any;
    checkOutDate: any;
    checkOutTime: any;
    petCodes: any;
    totalPrice: any;
    paymentType: any;

}
```

API for Creating Appointment

```
@RequestMapping(value = "/creatingAppointment", method = RequestMethod.POST)
public ResponseEntity<?> createAppointment(@RequestBody AppointmentDto appointmentDto)
{
    LinkedHashMap response = new LinkedHashMap();
    LinkedHashMap<String, Object> invoiceMap = new LinkedHashMap<>();

    try {
        Appointment appointment = new Appointment();
        BeanUtils.copyProperties(appointmentDto, appointment);

        Object objectAppointment =
commonService.checkTableEmpty("com.cba.hris.entity.Appointment", "Create
Appointment");

        if (null == objectAppointment) {
            appointment.setAppId(1);
            appointment.setAppCode("APPO-" + "1");
        } else {
            long appoId =
commonService.getLastRecordId("com.cba.hris.entity.Appointment", "appId", "Create
Appointment");
            appointment.setAppId(appoId + 1);
            appointment.setAppCode("APPO-" + (appoId + 1));
        }

        appointment.setCreatedBy("super");

        String date1 = convertDateToString(appointmentDto.getCheckInDate());
        String date2 = convertDateToString(appointmentDto.getCheckOutDate());

        invoiceMap.put("dates", date1 + " ->> " + date2);

        long noOfDays = 0;

        if(appointmentDto.getServiceCatCode().equals("Boarding")) {
            noOfDays = noOfDayCountBetweenTwoDays(date1, date2);
        }else{
            noOfDays = noOfDayCountBetweenTwoDays(date1, date2) + 1;
        }

        Service service = (Service)commonService.getObjectByColumnName("Service",
"serviceCode", appointmentDto.getServiceCode(), "Create Appointment");

        String strMap = String.valueOf(appointmentDto.getPetCodes()).replace("{",
"").replace("}", "")
            .replace("\\", "").replace("true", "").replace(", ", "");
        String[] petCodeArr = strMap.split(":");

        int noOfPets = petCodeArr.length;

        double totPrice =
noOfPets*noOfDays*Double.parseDouble(service.getServicePrice());
        appointment.setTotalPrice(String.valueOf(totPrice));

        Object object = commonService.addObject(appointment, "Appointment Creation");

        for(String petCode : petCodeArr){
            PetAppointment petAppointment = new PetAppointment();
            //BeanUtils.copyProperties(appointmentDto, appointment);
        }
    }
}
```

```

        Object objectPetAppointment =
commonService.checkTableEmpty("com.cba.hris.entity.PetAppointment", "Create
Appointment");

        if (null == objectPetAppointment) {
            petAppointment.setPetAppId(1);
            petAppointment.setPetAppCode("PETAPPO-" + "1");
        } else {
            long petAppoId =
commonService.getLastRecordId("com.cba.hris.entity.PetAppointment", "petAppId",
"Create Appointment");
            petAppointment.setPetAppId(petAppoId + 1);
            petAppointment.setPetAppCode("PETAPPO-" + (petAppoId + 1));
        }

        petAppointment.setPetCode(petCode);
        petAppointment.setAppCode(appointment.getAppCode());
        petAppointment.setCreatedBy("super");

        Object objectPetApp = commonService.addObject(petAppointment, "Appointment
Creation");
    }

    if (null != object) {

        //invoice
        invoiceMap.put("serviceName", service.getServiceName());

        invoiceMap.put("noOfPets", noOfPets);
        invoiceMap.put("noOfDays", noOfDays);
        invoiceMap.put("price", service.getServicePrice());
        invoiceMap.put("total",
noOfDays*noOfPets*Double.parseDouble(service.getServicePrice()));

        response.put("response", responseDto.getOk());
        response.put("message", "Appointment added Successfully.");
        response.put("invoice", invoiceMap);
        response.put("appointmentCode", appointment.getAppCode());
        return new ResponseEntity<>(response, HttpStatus.OK);

    } else {

        response.put("response", responseDto.getFail());
        response.put("message", "Error adding Appointment Details.");
        return new ResponseEntity<>(response, HttpStatus.OK);

    }

} catch (Exception e) {
    response.put("response", responseDto.getError());
    response.put("message", responseDto.getErrorMessage());
    return new ResponseEntity<>(response, HttpStatus.OK);
}
}
}

```

Appendix D - User Documentation

The system has been developed for a variety of users, including admin, customer, supplier, and veterinarian, and it provides a wide range of functionalities for these users. This manual instructs users on how to use system capabilities in order to meet management's requirements. Administrator has complete control over all modules. The documentation for the user with administrative privileges is compiled by detailing all of the system's module functions.

Getting familiar with the system

System was built in user friendly understandable interface views. Proper flow include in each module. Typing the URL in the address bar navigates to the Login page (Figure D.1).

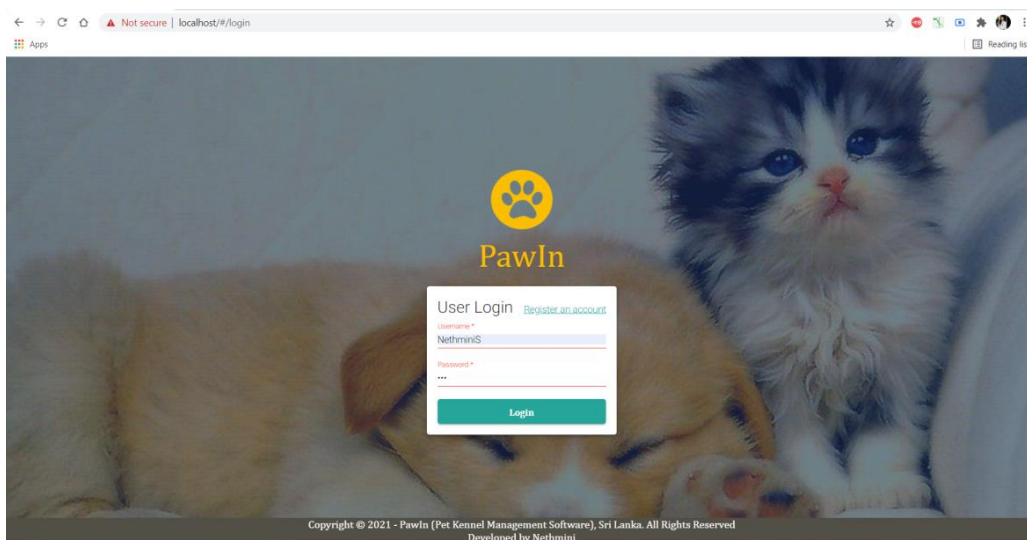


Figure D. 1: Login page

For the purpose of gain authorized access to the system, the user should provide correct username and password. After logged in, the user is directed to the home page.

After logging in, the dashboard main page appears. The modules can be found in the left side navigation bar's menu bar drop down. And right side top user log name navigation display log out and user profile view module. Figure D.2 displays login user privileges and Figure D.3 displays navigation bar for main modules.

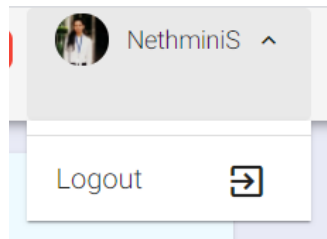


Figure D. 2: Display login

user privilege

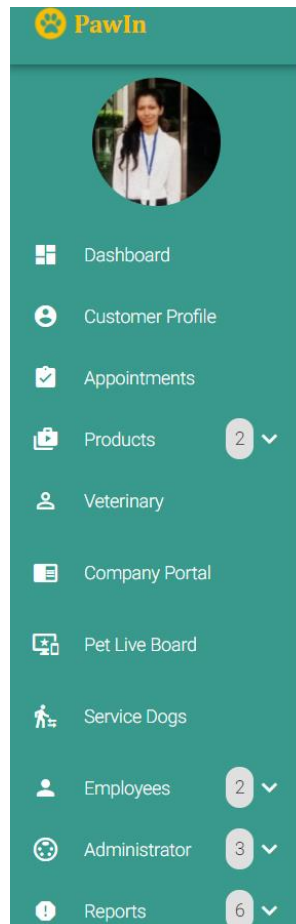


Figure D. 3: Navigation bar

General Guidelines in Using Common Form Controls

Via button click, complete the appropriate function named by each button. Figure D.4 shows common buttons used in the system.

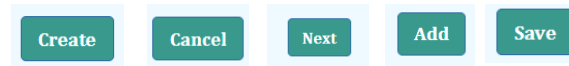


Figure D. 4: Common buttons

Common calendar display using reusable java script. Figure D.5 display date picker using system.

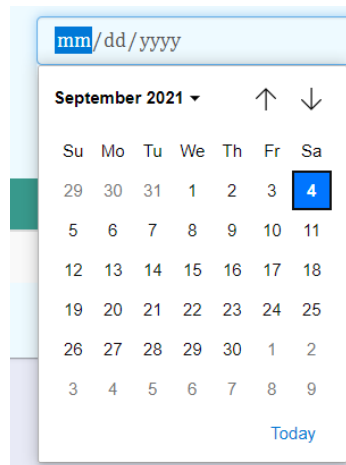


Figure D. 5: Date picker using system

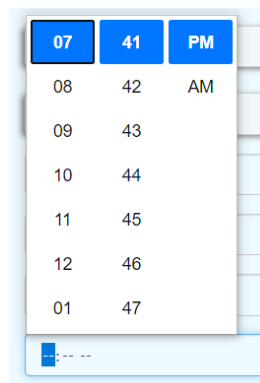


Figure D. 6: Time picker using system

General grid view use for all data search according to given filtrations. Figure C.7 show common grid view used in modules.

















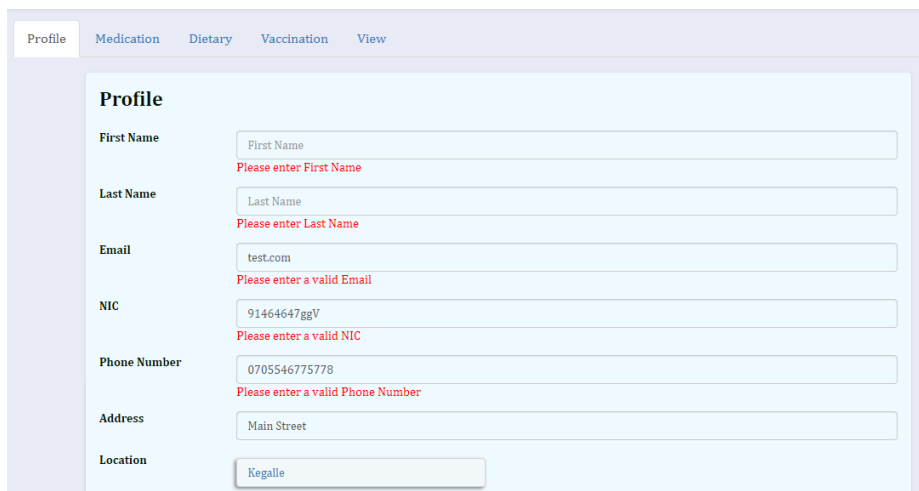
Resource Name	Max Weight	Is Clean Needed	Action
Superior	200	No	  
Dulax	150	Yes	  
Winter	100	No	  
Premium	500	No	  
Otum	1000	Yes	  
Deak	300	No	  

Figure D. 7: Common grid view used in modules

Validations were added to popup when save or edit details without main functions. Figure C.8 display error messages in item creation view save without mandatory fields.



The screenshot shows a 'Profile' form with the following fields and error messages:

- First Name:** Field contains 'First Name'. Error message: 'Please enter First Name'.
- Last Name:** Field contains 'Last Name'. Error message: 'Please enter Last Name'.
- Email:** Field contains 'test.com'. Error message: 'Please enter a valid Email'.
- NIC:** Field contains '91464647ggV'. Error message: 'Please enter a valid NIC'.
- Phone Number:** Field contains '0705546775778'. Error message: 'Please enter a valid Phone Number'.
- Address:** Field contains 'Main Street'.
- Location:** Field contains 'Kegalle'.

Figure D. 8: Display error messages

When save with correct details and fill all mandatory fields successfully save message display (Figure D.9) and detail view shown.

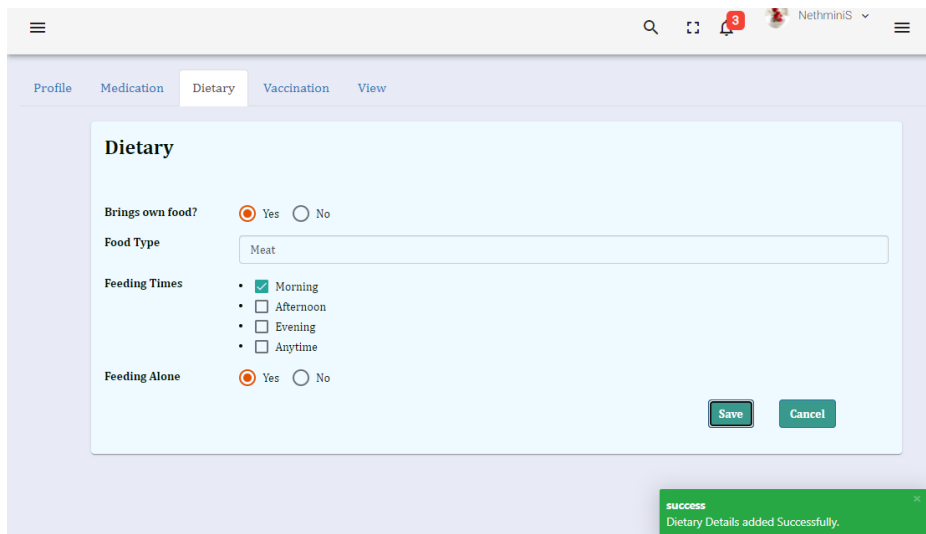


Figure D. 9: Save message display

Modules

The sections below provide brief summaries of some of the most important modules and features. The general recommendations, which are the same for all of the system's module capabilities, explain how to update and delete data. As a result, only the unique characteristics are discussed further below.

- Create Customer Profile

Customer /Admin can navigate to customer profile by clicking on “Customer Profile” in side bar. By entering customer first name, last name, email, address, phone number and location customer profile can be created. Then pets can be added to particular customer profiles. Customer can enter medication, dietary, vaccination, details by selecting added pets. Figure D. 10 to D.11 show the process of creating a customer profile, adding pets and adding dietary/vaccination/medication details.


Profile Medication Dietary Vaccination View

Profile

First Name
Last Name
Email
NIC
Phone Number
Address
Location

Figure D. 10: Create Customer Profile

Add Pet



Pet Name

Birth Date

Type Dog Cat

Pet Weight kg

Gender Male Female

Breed

Color

Figure D. 11: Add Pet

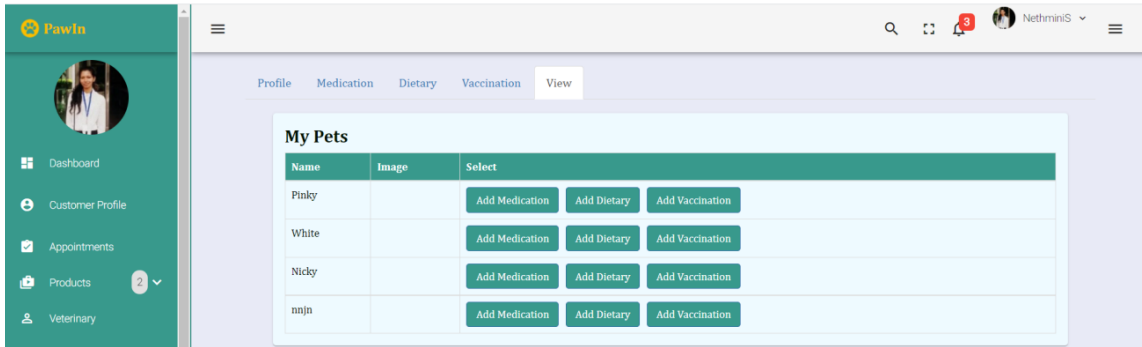


Figure D. 12: Add Medication/Add Dietary/Add Vaccination by selecting pets

- Company Portal

Administrator/Customer can navigate to Company Portal and has the ability to see pets which are in waiting, check in and checkout statuses by selecting the desired date and search. Customer can only see his own pets while administrator can see all the pets. Figure D.13 depicts Company Portal for selected date.

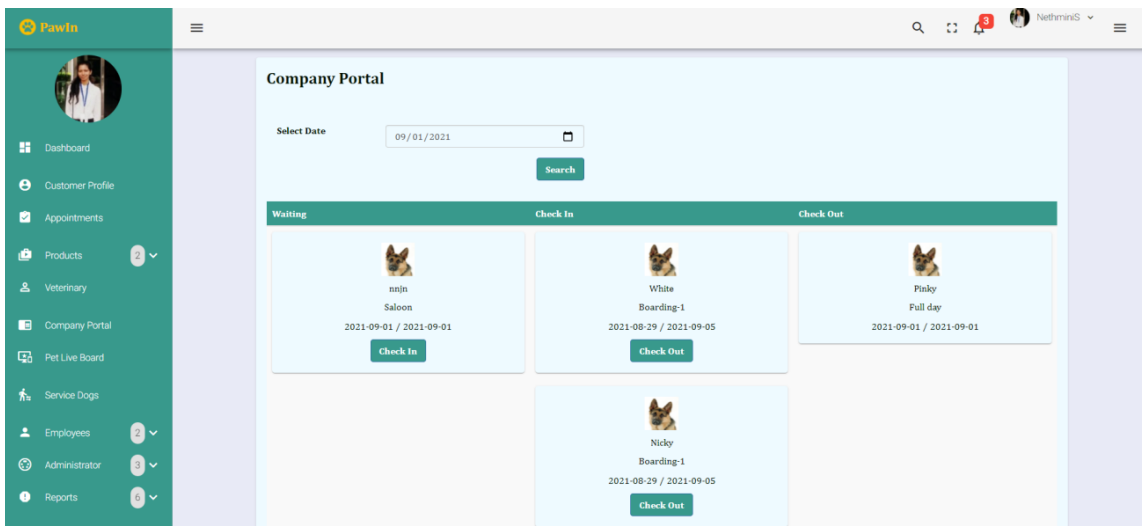


Figure D. 13: Company Portal

- Pet Live Board

By navigating Pet Live Board administrator has the ability to get pets which are checked in for the current date. Daycare and boarding service categories would be considered and administrator can filter pets for only daycare, only boarding or both daycare and boarding (all) by clicking particular buttons. Figure D.14 represents Pet Live Board for the current date.

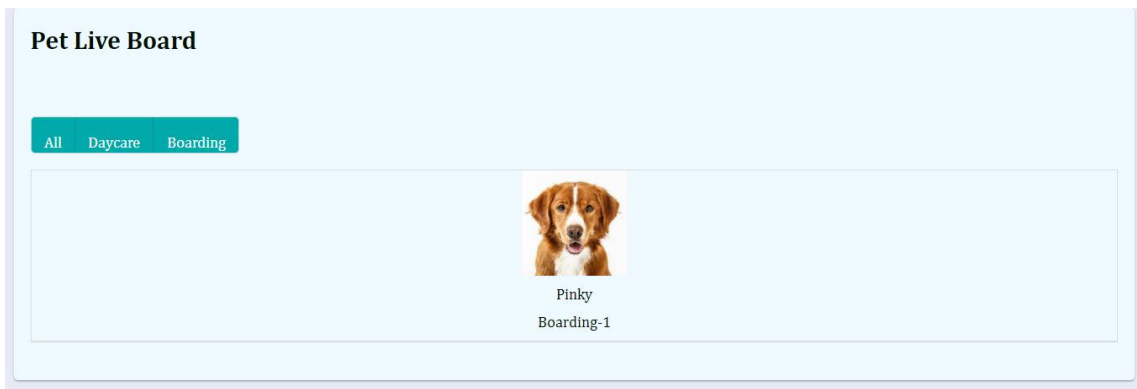


Figure D. 14: Pet Live Board

- Employee Schedule

Administrator can create schedules for employees by selecting particular employees. Start time and end time of days can be added for selected employee. Figure D.15 to D.17 show Employee Schedule after searching an employee.

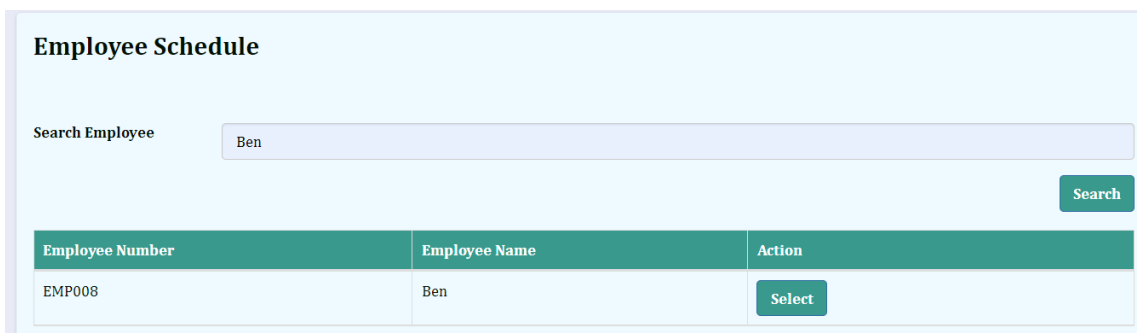


Figure D.15: Search Employee

Add Schedule ✕

Tuesday

Wednesday

Thursday

Friday

Saturday

Start Time 🕒

End Time 🕒

Add
Cancel

Figure D. 16: Add Schedule for selected employee

Day	Start Time	End Time	Action
SUNDAY	09:00	18:00	✎ ✓ 🗑️
MONDAY	07:00	18:00	✎ ✓ 🗑️
TUESDAY	10:00	19:00	✎ ✓ 🗑️
THURSDAY	09:00	17:00	✎ ✓ 🗑️
FRIDAY	09:00	17:00	✎ ✓ 🗑️

Figure D.17: View employee schedule for selected employee

Appendix E- User Evaluation Results

The questionnaire has been accessed by ten pet institutions and similar domains in island wise. Responses for some of the questions in the questionnaire can be presented graphically as follows:

1. Do you like to recommend this software to your friend or family member?

Most of the participants of this survey have selected the 7th rate for software recommendation while no low rate responses. Hence, PawIn software can be recommended for the people who need to maintain pet in house or kennel management. Figure E.1 depicts number of responses given for each rate starting from 1 to 10.

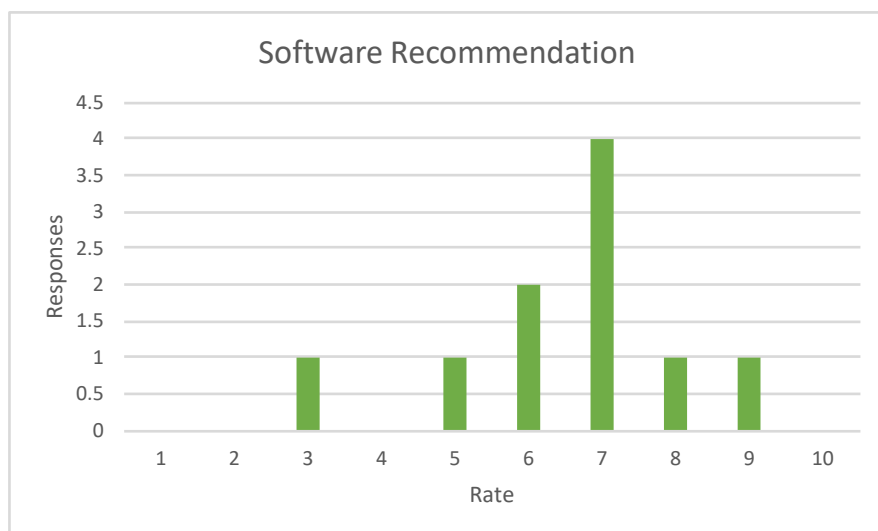


Figure E. 1: Responses for Software Recommendation

2. How satisfied are you with the software's ease of use?

Ease of use for users is one of the major criteria for measurement of good software. 40% participants have marked as somewhat satisfied with the ease of use of the system. Following Figure E: 2 illustrates the responses for particular question from participants.

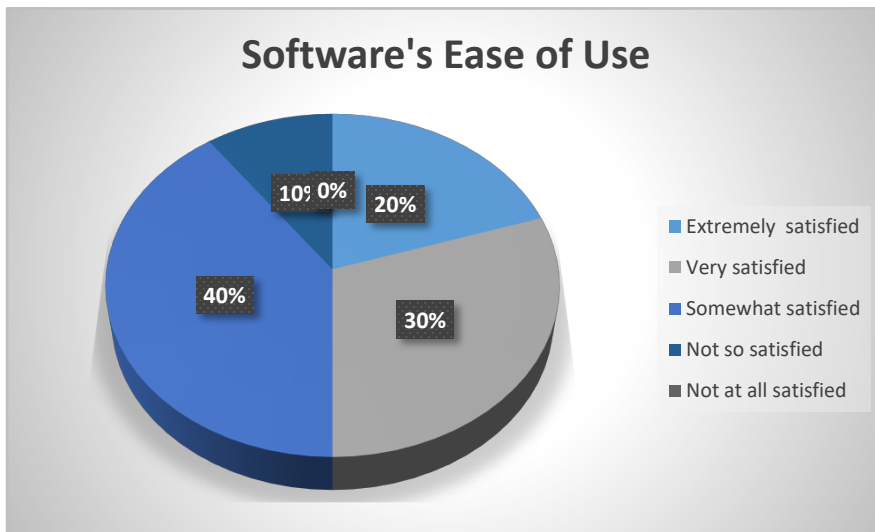


Figure E. 2: Responses for Software’s Ease of Use

3. Are you satisfied with the features of the software?

More than half from the participants have responded as very satisfied or somewhat satisfied with the features of the limited version of PawIn software. Therefore, the researcher can proceed with full version of the software for the purpose of marketing and selling. Figure E.3 shows satisfactory level against number of responses.

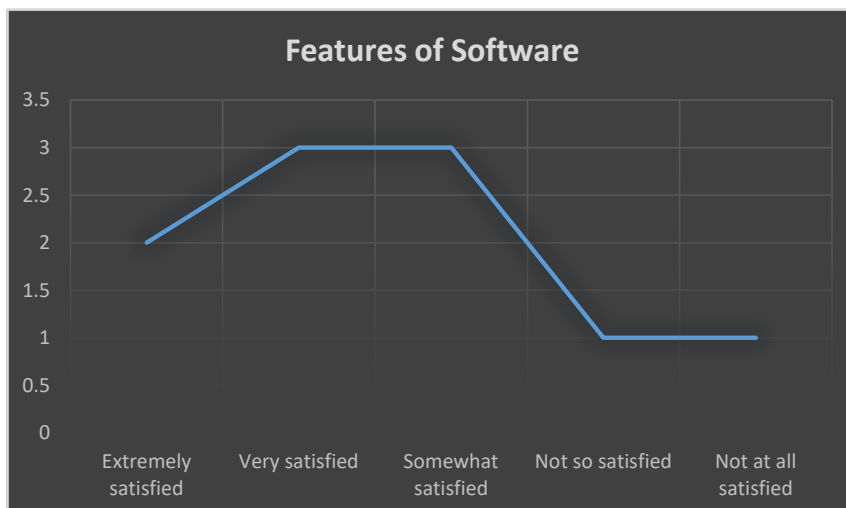


Figure E. 3: Responses for Features of Software