PAWS' OWN

PET CARE PRODUCTS AND ACCESSORIES MANAGEMENT SYSTEM

Project Report Submitted by

Lakshmi Sunil

Reg. No.: AJC18MCA-I042

In Partial fulfillment for the Award of the Degree of

INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2022-2023

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**PAWS' OWN"** is the bonafide work of **LAKSHMI SUNIL (Regno: AJC18MCA-I042)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

Dr. Bijimol T. K. **Ms. Meera Rose Mathew**

Internal Guide **Coordinator**

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department **External Examiner**

DECLARATION

I hereby declare that the project report **"PAWS' OWN - PET CARE PRODUCTS AND ACCESSORIES MANAGEMENT SYSTEM"** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (INT MCA) from APJ Abdul Kalam Technological University, during the academic year 2022-2023.

Date: 12/04/2023 **LAKSHMI SUNIL**

Kanjirappally **Reg:. AJC18MCA-I042**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew**, Assistant Professor for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Dr. Bijimol T. K.,** Assistant Professor for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

LAKSHMI SUNIL

ABSTRACT

PAWS' OWN focuses on developing a web application where we can purchase a variety of pet care products and accessories. The proposed system is exclusively for Dogs and Cats. A Pet care products and accessories store is a retail business which sells pet care resources to the public. A variety of animal supplies and pet accessories are also sold in pet care shops. People should be able to come to know about the daily routine and proper care they should give to their pets. In manual system, people visit the shop and from the available products, they choose and buy the items they want and proceed with the payment. The local shops are time-consuming and have only a limited number of products. The proposed system computerizes the marketing of pet care products. It also reduces human efforts, saves time and other resources, and provides a wide range of products for pets. It also has options to book appointments with the veterinary doctor and can book vaccinations for pets at the comfort of their home. The system is designed for the physical shops that helps the customers to purchase goods via the internet. The main modules of the system are Admin module, Customer module, Veterinarian module and Vaccine Center module. Admin functionalities include products management, customer management, veterinarian management, feedback management, stock management, revenue prediction, sentiment analysis and the admin control the entire system. Customer functionalities include browsing of products, information regarding pet health schedule, pet vaccination schedule, booking appointments, booking vaccinations, view orders, voice search, retrieval of pet information using QR code technology, view appointment history, vaccination history, vaccine reporting and payment. The system also provides the customers an AI based chatbot that can resolve the queries submitted by the customers. Functionalities of Veterinarian include managing profile, managing appointments, giving prescriptions, and managing pet health care schedule. Functionalities of vaccine center include adding vaccines, adding vaccine agency, updating vaccine information, stock management and blocking vaccines or vaccine agencies. The proposed system manages all the activities that are carried out in manual shops and maintains a good customer relationship.

CONTENT

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language
CSS	-	Cascading Style Sheet
AJAX	-	Asynchronous JavaScript and XML
SQL	-	Structured Query Language
UML	-	Unified Modeling Language
PHP	-	Hypertext preprocessor

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Pet care products and accessories management system is a web application used to oversee the operations of a physical pet store. The suggested approach meets all the essential requirements for pet owners to give their animals the finest care possible. The web-based application for making purchases is this one. This system offers the ability to shop for pet supplies while simultaneously displaying all available information about the store. In order to schedule consultation appointments with the veterinarian to discuss the health of their clients' dogs, it also offers an online application form. The pet owners can schedule vaccination appointments from the convenience of their own homes.

1.2 PROJECT SPECIFICATION

Pet care products and accessories management system is a web-based application that allows pet owners to make purchases and request services. Customers can make an appointment with the veterinarian and schedule vaccines using the online application form. Customers have the option to search or browse a selection of products thanks to the system. Additionally, it offers the vaccination plan and pet health care schedule that will help the consumers keep their pets' good health. The web-based system is crucial in order to save customer data and offer a better, more organized way of conducting everyday business for the pet care products and accessories store. The clients will experience a simple and easy accessibility to all information of system. The system includes 4 modules. They are:

1. Admin Module

The site admin has an overall control on the website. The admin’s functionalities include:

- Login to the system: Admin can log in the system with valid username and password
- Add/View/Update/Delete Products: The admin has the functionalities which includes adding new products, updating products and removing products.
- Update/Activate/Deactivate Veterinarian: Admin can update vet details, block or unblock veterinarian.
- Activate/Deactivate Customers: Admin can block or unblock the customers
- Stock Management: Admin can update the stock of products
- Update order status: Admin can change the order status of products ordered by the customers including shipping status and delivery status.
- Sales Prediction: Admin can see the predicted revenue in the upcoming years based on

the past years.

- View Feedbacks from customers: Admin can read the feedbacks submitted by customers.
- Sentiment Analysis: The feedbacks submitted by the customers can categorized based on positive, negative, and neutral comments. The system provides a graphical representation of feedbacks for the admin.
- Order Statistics: Admin can view order details specific to a month or a year.
- Report generation: Admin can generate reports of orders, products, customers, pets registered, consultations and vaccinations.

2. Customer Module

Customers are the main users of the system. Their functionalities include:

- Registration and Login: A new user can register to the system with valid user credentials.
- Update Profile/Password: The customer can update their profile, change password and can recover the password.
- Register Pets: The customers can register their pets on the website for vaccination bookings and for consultation with the vet.
- View and update pet information: The customers can view the registered pets and also can update their information.
- Pet card generation: The customers can generate QR code for each of their registered pets that can be used as pet tag for identifying them if pets are lost.
- Book and view consultations: The customer can book appointments with the veterinarian for various pet health concerns and can view consultation history and upcoming consultations.
- Book and view Vaccinations: The customer can book vaccinations for their pets and can view vaccination history and upcoming vaccinations.
- Submit report: Customers can report a vaccine to the vaccine center in case there is any undesirable symptoms are exposed by the pets post vaccination.
- Prescription: Customers can download prescriptions submitted by vet which suggests the medicine to be given to the pets.
- Regular Search and Voice Search: Search functionality is given to the customers to search products, pets, consultations, and vaccinations.
- Add to cart and purchase products: Customers can add the products that they wish to buy to the cart.
- Purchase Products: The customers can check out the products in cart and pay using Razor pay payment gateway.
- Cancel Orders: Customers can cancel the order before delivery

- Update Address: Customers can update their address
- View orders: The customers can view order history and can see current status of the order including shipping and delivery status.
- Submit Feedbacks: The customer can send the feedbacks to the admin of the system regarding the products and services.
- View Pet Health Schedule/ Vaccination Schedule: The customer gets information about the pet health schedule and vaccination schedule.
- Personal Assistant: The web application provides an AI based chatbot to resolve the real-time queries of customers.
- Account Deletion: Customers can delete their account if they want.

3. Veterinarian

The functionalities of veterinarian include:

- Login to the system: The veterinarian can log in to the system with the valid user credentials.
- View/Update Profile: The veterinarian has the functionality to update their personal details
- View Consultations: The veterinarian can view appointment history, upcoming appointments, and has the option to view today's appointments.
- Approve/Decline appointments: The veterinarian can approve the consultations and can also deny them in case the doctor is not available at the booked slot.
- Add and Edit Prescription: Vet can suggest medicines and can give prescription for consultations that are completed.
- View/Add/Update Pet health Schedule: The veterinarian has the functionality to add, update and delete the pet health schedule.

4. Vaccine Center

The functionalities of vaccine center include:

- Login to the system: The vaccine center can log in to the system with the valid user credentials.
- Add/View/Update Vaccine Info: The vaccine center has the functionality to add and update vaccine information.
- Update vaccine availability: The vaccine center can update vaccine stock.
- Add/View/Update Vaccine Agency: The vaccine center can add, view and can update vaccine agency information.
- Approve/Deny Bookings: The vaccine center has the functionality to approve bookings

submitted by the customers

- View and download vaccination statistics: The vaccine center can view the number of vaccinations done, not done and to be done. It also includes the functionality to download vaccination report and vaccination data specific to each pet registered.
- Report Vaccine or Vaccine Agency: The vaccine center can report the vaccine or can block the vaccine agency based on whether or not any report is submitted by the customer.
- View Bookings: The vaccine center can view booking history, vaccinations to be done, and has the option to view today's vaccination bookings.
- Data Visualization: Vaccine center can see a graphical representation of vaccinated and non- vaccinated counts.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System study is the process of examining data, pinpointing problems, and using the information to recommend system upgrades. In order to effectively accomplish a procedures or business's goals and purposes, systems and processes must be developed. It necessitates close coordination between system developers and users. The system is extensively analyzed, the inputs are acknowledged, and the system is viewed as a whole in order to pinpoint the problem areas. As a suggestion, the solutions are provided. The idea is evaluated and the necessary changes are made in response to user requests. System analysis is the process of acquiring and analyzing data, diagnosing issues, and using the data to suggest system changes. The system users and system developers must communicate extensively during this problem-solving process. Any system development process should start with a system analysis or research. The system is meticulously examined and analyzed. The system analyst assumes the role of an interrogator and delves deeply into how the current system functions. The input to the system is identified, and the system is seen as a whole. The various procedures can be linked to the outputs from the organizations.

Understanding the issue, identifying the pertinent and important variables, analyzing and synthesizing the different elements, and selecting the best or, at the very least, most acceptable course of action are all part of system analysis. The process must be thoroughly studied using a variety of methodologies, including questionnaires and interviews. To reach a conclusion, the information gathered by these sources must be carefully examined. Understanding how the system works is the conclusion. The current system is the name of this system. Now, the current system is carefully examined, and issue areas are found. The designer now acts as a problem-solver and works to resolve the issues the business is having. Proposals are made in place of the solutions. The proposal is then analytically compared to the current system, and the best one is chosen. The user is given the opportunity to approve or reject the suggestion. On user request, the proposal is assessed and appropriate revisions are made. As soon as the user is content with the suggestion, this loop breaks. The process of acquiring and analyzing data in order to use it for future system studies is known as preliminary study. Initial research is a problem-solving activity that necessitates close coordination between system users and developers. It conducts a number of feasibility studies. These investigations provide an approximate estimate of the system activities, which can be used to determine the tactics to be used for an efficient system research and analysis.

2.2 EXISTING SYSTEM

Under the current method, a customer must physically visit the store to select products for their dogs. The method takes time and is not user-friendly because it calls for a manual visit from a human. The consumer must be informed about the product before making a purchase decision. If not, it will be challenging to distinguish the product from the many others that are accessible that have similar appearances.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Insufficient awareness of goods and services.
- Customers have to be physically present at the shop before choosing any item for purchase.
- Lack of adequate storage facility
- Difficulty in transaction process
- A time-consuming process, being in queue for visiting vets and for vaccinations.
- Less user friendly

2.4 PROPOSED SYSTEM

The main goal of PAWS' OWN is to create an online application where users can buy various pet case products and accessories without having to physically visit a store. The suggested system differs from the current system in that it is specifically created for dogs and cats. The system administrator has control over both people and goods. Customers of the system can do product searches across a range of categories, add items to shopping carts, make purchases, view order histories, schedule appointments, order vaccinations, view appointment and vaccination histories, report vaccines, and view pet health and vaccination schedules. Through the app, a consumer can place an order for the item they want to purchase. The technology enables customers to discover the best items for their dogs and learn how to properly care for them.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Online shopping is possible from anywhere at anytime
- Customers get a wide range of products to choose from
- Online payment and delivery of products to doorstep
- No pressure, no crowed, simply convenient
- Online booking of consultations and vaccinations

CHAPTER 3

REQUIREMENT ANALYSIS

4.1 FEASIBILITY STUDY

A feasibility study is carried out to ascertain whether the project will, after completion, achieve the organization's goals in relation to the work, effort, and time put into it. The developer can predict the project's future and usefulness thanks to a feasibility assessment. A feasibility study's foundation is the system proposal's viability, which takes into account the system's impact on the organization, ability to meet user needs, and effective use of resources. So, before a new application is accepted for development, it typically goes through a feasibility study when it is presented. The document outlines the project's viability and contains a number of factors that were carefully taken into account throughout this project's feasibility study, including its technical, economic, and operational viabilities. It has the following features

4.1.1 Economic Feasibility

Analyses of costs and benefits are necessary to support the developing system. Criteria to ensure that the project that would produce the best outcomes the quickest is given priority. One of the variables is the cost associated with establishing a new system. Some significant financial queries raised during the initial probe include the following:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

There are no manual costs involved with the suggested system because it was developed as part of a project. Additionally, the fact that all of the resources are already at hand indicates that the system may be developed affordably. System costs, development costs, and hosting costs make up the three cost categories for the suggested system. All calculations show that the project was developed at a low cost because open-source software was used during the whole development cycle. Utilizing cost-effective resources, the system was created.

4.1.2 Technical Feasibility

The system needs to be assessed first from a technical standpoint. An overview design of the system requirement in terms of input, output, programs, and procedures must serve as the basis for the assessment of this feasibility. The inquiry must next advise the kind of equipment, necessary procedure for constructing the system, and means of operating the system once it has been designed after having identified an outline system.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be created in such a way that the necessary functionality and performance are met within the constraints. The fact that a newer version of the same software still functions with an older version means that the system can still be used even though the technology may become outmoded with time. There are therefore few restrictions associated with this undertaking. The system's front end and back end have both been constructed using HTML and CSS, and the project can technically be developed. The system, which has an Intel i7 core processor, 8GB of RAM, and a 1TB hard drive, performed well as well.

4.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be advantageous because, when created and implemented, it would achieve the goals. The project is deemed to be behaviorally feasible after carefully weighing all behavioral factors. PAWS' OWN, GUI is user friendly so that users can easily use it without any training.

3.1.4 QUESTIONNARIE

1. What are the goals and objectives of Online Shopping System?
   The primary goal of the proposed system is to reach maximum customers at the right time to increase sales and profitability of the business.

2. Who are the users of the system?
   The users of the system are admin, veterinarian and the customers.

3. Who are the customers of online pet stores?
   Pet-Specific Customers, includes young pet owners and families centered on their pets.

4. How are dogs and cats vaccinated?
   The current manufacturer's recommendation is for use of this vaccine from 8 weeks of age.

Administer one dose at 6–8 weeks and one dose at 10–12 weeks of age. Two doses 2–4 weeks apart. Annually or more often in very high-risk animals not protected by annual booster.

5. Which are the human foods you can feed your dogs and cats?

   It includes Salmon, Bread, Eggs, Spinach, Chicken, Turkey, Green beans, Broccoli

6. What is the frequency of sales in the proposed system?

   The frequency of sales depends upon the need of the pet owners.

7. How the proposed system helps the customers in finding the best products?

   The proposed system enables the customer to browse the firm's range of products and services, view photos or images of the products, along with information about the product specifications, features and prices.

8. Is online consultation for pets good?

   Online consultation is beneficial if your condition limits your mobility or perhaps you don't have access to transportation. 80% of health concerns can be solved through online consultations. Whatever the reason, you can get medical help from the comfort of your own home.

9. What is the need for an online shopping system and how it overcomes the disadvantages of the manual system?

   An online shopping system saves time and efforts. The convenience of shopping at home is a significant feature of online shopping. Wide variety/range of products are available. There are good discounts or lower prices are provided to the customers.

10. How to make an order in an online shopping system?

    First of all, we are logging on to the system. Then we browse or search products. After finding the products that matches your need is added to the cart. Then the items in the cart are proceeded to checkout. The delivery address is provided and the user then reaches the payment user interface where the user can make payment by using different methods.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel Core i7

RAM - 8 G B

Hard disk - 1 T B

3.2.2 Software Specification

Front End - HTML, CSS

Backend - MYSQL

Client on PC - Windows 11

Technologies used - JavaScript, HTML5, AJAX, jQuery, PHP, CSS, Machine Learning

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is an abbreviation for PHP: Hypertext Preprocessor. PHP is a server-side scripting language made especially for creating websites. Since it is open-source, you can use and download it for nothing. Learning and using it are quite simple. ".php" is the extension of the files. Both the original PHP and later variants were inspired by Rasmus Lerdorf. A compiler is not necessary because it is an interpreted language. Instead of contacting an external file to handle data, PHP commands can be directly included into an HTML source document. Because of the limitations on how it may be used, it has also developed to include a command-line interface capability and is incompatible with the GNU General Public License (GPL). Most web servers support the free deployment of PHP, which is also available as a standalone shell on practically all platforms and operating systems. Numerous databases, including Oracle, Microsoft SQL Server, MySQL, PostgreSQL, Sybase, and Informix, can be combined with it. It can be used to restrict user access and is strong enough to support a content management system like WordPress. PHP codes are performed on the server, as opposed to HTML codes, which are directly rendered on the browser, which is what sets PHP apart from client-side languages like HTML.

3.3.2 MySQL

Oracle Corporation created, distributed, and provided support for MySQL, the most well-known Open-Source SQL database management system. The most recent details regarding MySQL software are available on the MySQL website. A popular relational database management system (RDBMS) is MySQL. The best option for both small and large applications is MySQL, which is free.

- A database management system is MySQL: A systematic collection of data is called a database. It might be anything, such as a straightforward grocery list, a photo gallery, or the enormous amount of data in a business network. A database management system, such as MySQL Server, is required to add, access, and process data contained in a computer database. Database management systems, whether used as stand-alone programmes or as a component of other programmes, are essential to computing because computers are excellent at processing vast volumes of data.

- A relational database, like the ones used by MySQL, divides up data into individual tables rather than storing it all in one place. Physical files that are optimized for speed contain the database structures. The logical model provides a flexible programming environment with objects like databases, tables, views, rows, and columns.

- The MySQL software is open source, which allows for anyone to use and alter the programme. The MySQL software is available for free download and usage online by anyone. You are free to examine the source code and modify it as necessary. The GPL (GNU General Public License) is used by the MySQL software to specify what you are allowed to do and are not allowed to do with the software in certain circumstances. You can purchase a commercially licensed version from us if the GPL makes you uncomfortable or if you need to integrate MySQL code into a for-profit application.

- The MySQL Database Software is a client/server system that includes a multi-threaded SQL server that supports many back ends, a number of unique client programmes and libraries, administrative tools, and a broad variety of application programming interfaces (APIs). Additionally, we provide MySQL Server as an integrated multi-threaded library that you can link into your programme to create a standalone offering that is smaller, faster, and simpler to operate

.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step in the development of any engineered system or product. A creative process is design. The secret to an efficient system is a decent design. The process of using different methodologies and concepts to specify a process or a system in enough detail to allow for its physical realization is referred to as "design." One way to describe it is as the process of using different methodologies and concepts to specify a device, a process, or a system in enough detail to allow for its physical realization. Regardless of the development paradigm that is employed, software design forms the technical core of the software engineering process. The architectural detail needed to construct a system or product is developed through the system design. This programme has also through the best possible design phase, fine tuning all efficiency, performance, and accuracy levels, as in the case of any systematic technique. A user-oriented document is converted into a document for programmers or database staff throughout the design phase. The two stages of system design development are logical design and physical design.

4.2 UML DIAGRAM

A common language known as UML is used to design, visualize, build, and document the software system artefacts. It was developed by the Object Management Group (OMG), and the OMG received a draught of the UML 1.0 definition in January 1997. Unified Modelling Language is known as UML. It is a graphical language that is used to create software blueprints, unlike other popular programming languages like C++, Java, COBOL, etc. A general-purpose visual modelling language for software system visualization, specification, construction, and documentation is what UML is known as. UML is not just used to represent software systems, despite the fact that this is its most common application. It is also used to model systems that are not software-based. For instance, the manufacturing facility's process flow, etc. Although UML is not a programming language, tools can be used to generate code using UML diagrams in a variety of languages. The analysis and design of objects-oriented systems are directly related to UML. UML has been standardized to the point where it is now an OMG standard. A comprehensive UML diagram that depicts a system is made up of all the elements and relationships. The most crucial aspect of the entire procedure is the UML diagram's aesthetic impact. It is completed by using all the additional components. The following nine diagrams are part of UML:

- Use case diagram
- Sequence diagram
- State chart diagram
- Activity diagram
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram
- Collaboration diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between system components. A process for identifying, outlining, and organizing system needs is called a use case. The word "system" here refers to a thing that is being created or run, like a website for mail-order product sales and services. UML (Unified Modelling Language), a standard notation for the modelling of real-world objects and systems, uses use case diagrams. The planning of general requirements, the validation of a hardware design, the testing and debugging of a software product in development, the creation of an online help reference, or the completion of a job focused on customer support are all examples of system objectives. Use cases in a setting where products are sold, for instance, may be item ordering, catalogue updating, payment processing, and customer relations. A use case diagram contains four components:

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

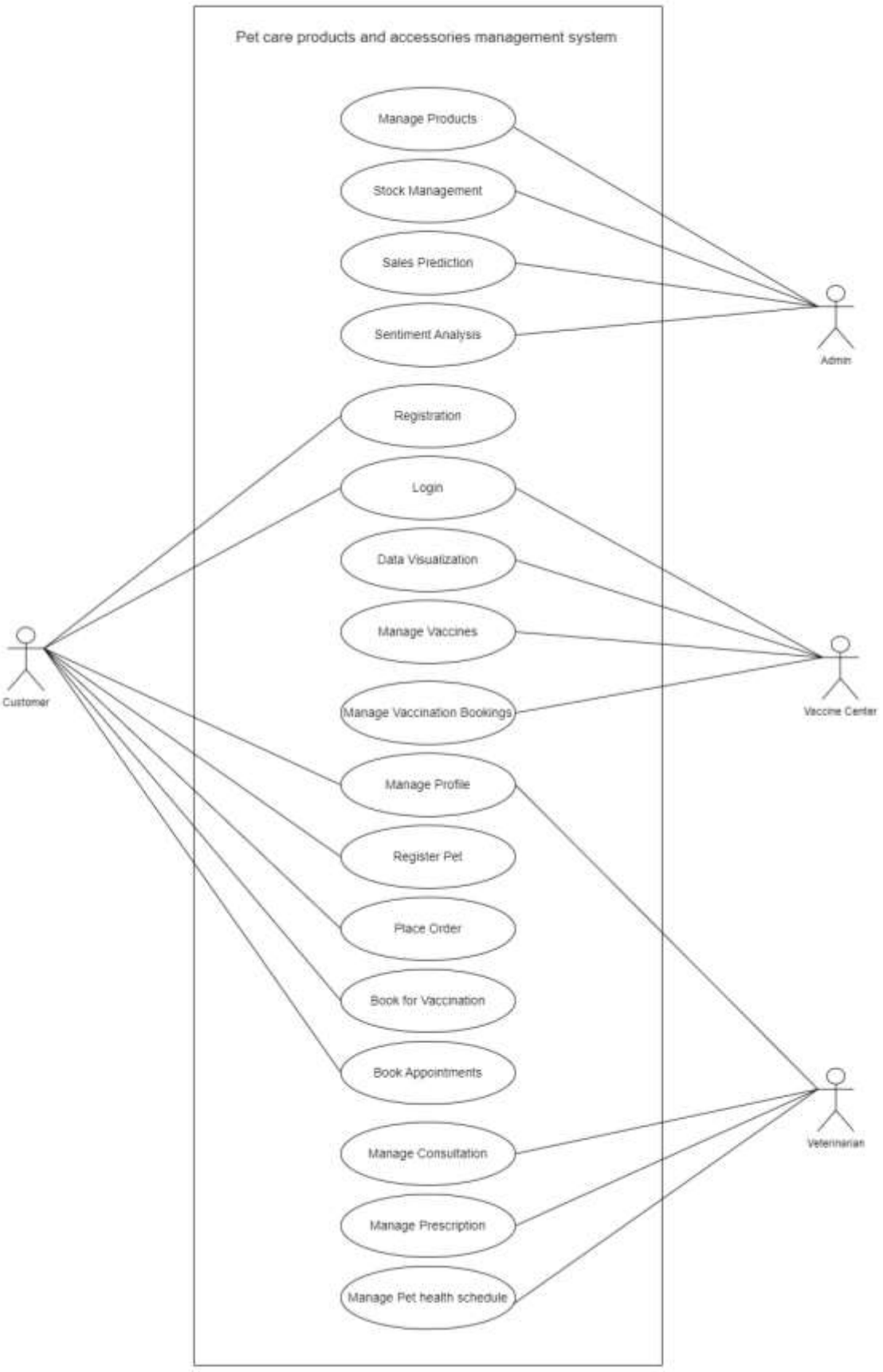


Figure 4.2.1.1: Use Case Diagram

4.2.2 SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that demonstrates the order and interactions of several things. It is an implementation of a message sequence diagram. Object interactions are arranged in temporal sequence in a sequence diagram. It shows the classes and objects involved in the scenario as well as the flow of messages that must be exchanged for the objects to work as intended. In the Logical View of the system being developed, sequence diagrams are often connected to use case realizations. Event diagrams and event scenarios are other names for sequence diagrams. A sequence diagram displays various concurrent processes or objects as parallel vertical lines (lifelines), and the messages passed between them as horizontal arrows, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

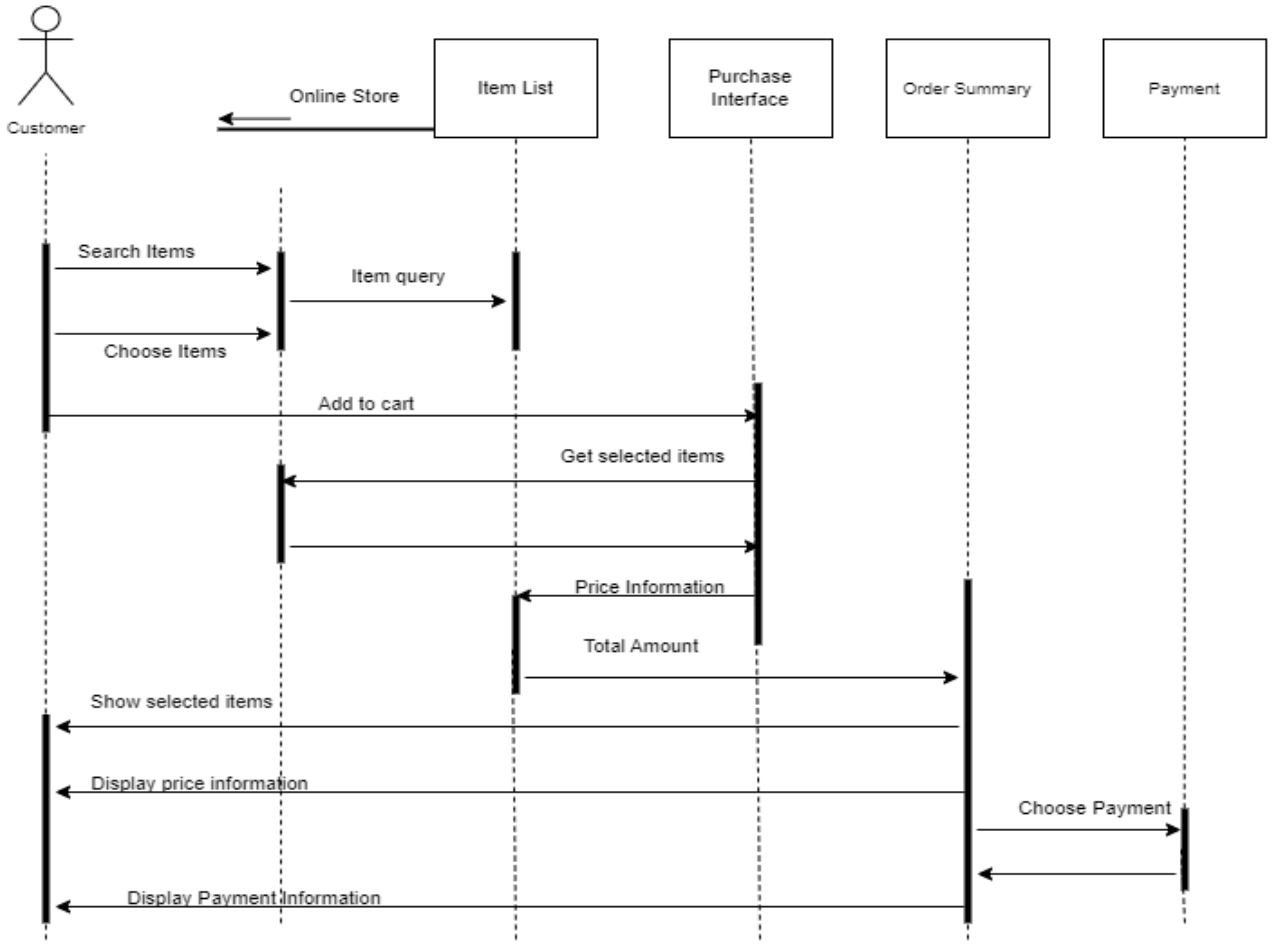


Figure 4.2.2.1: Sequence diagram for the proposed system

4.2.3 State Chart Diagram

It describes various system components' statuses. The states are unique to a particular system object or component. A state machine is depicted in a state chart diagram. A state machine is a device that distinguishes between various states of an object and controls these states in response to internal or external events. Throughout an object's existence, they define several states, and these states are altered by events. Diagrams of state charts can be used to model reactive systems. A system that reacts to internal or external events is known as a reactive system. The transfer of control between states is shown in a state chart graphic. States are described as a situation where an object existing and changes as a result of an event. Modelling an object's lifetime from conception to termination is the primary goal of a state chart diagram. For both forward and reverse engineering of a system, state chart diagrams are also employed. The reactive system's modelling is the primary goal, though.

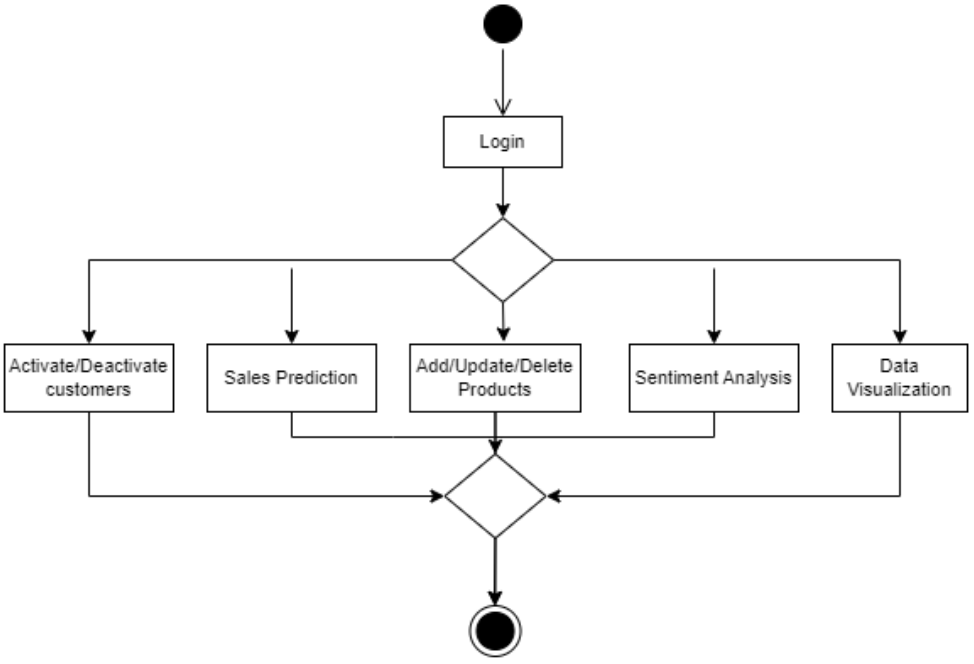


Figure 4.2.3.1: State Chart Diagram for Admin

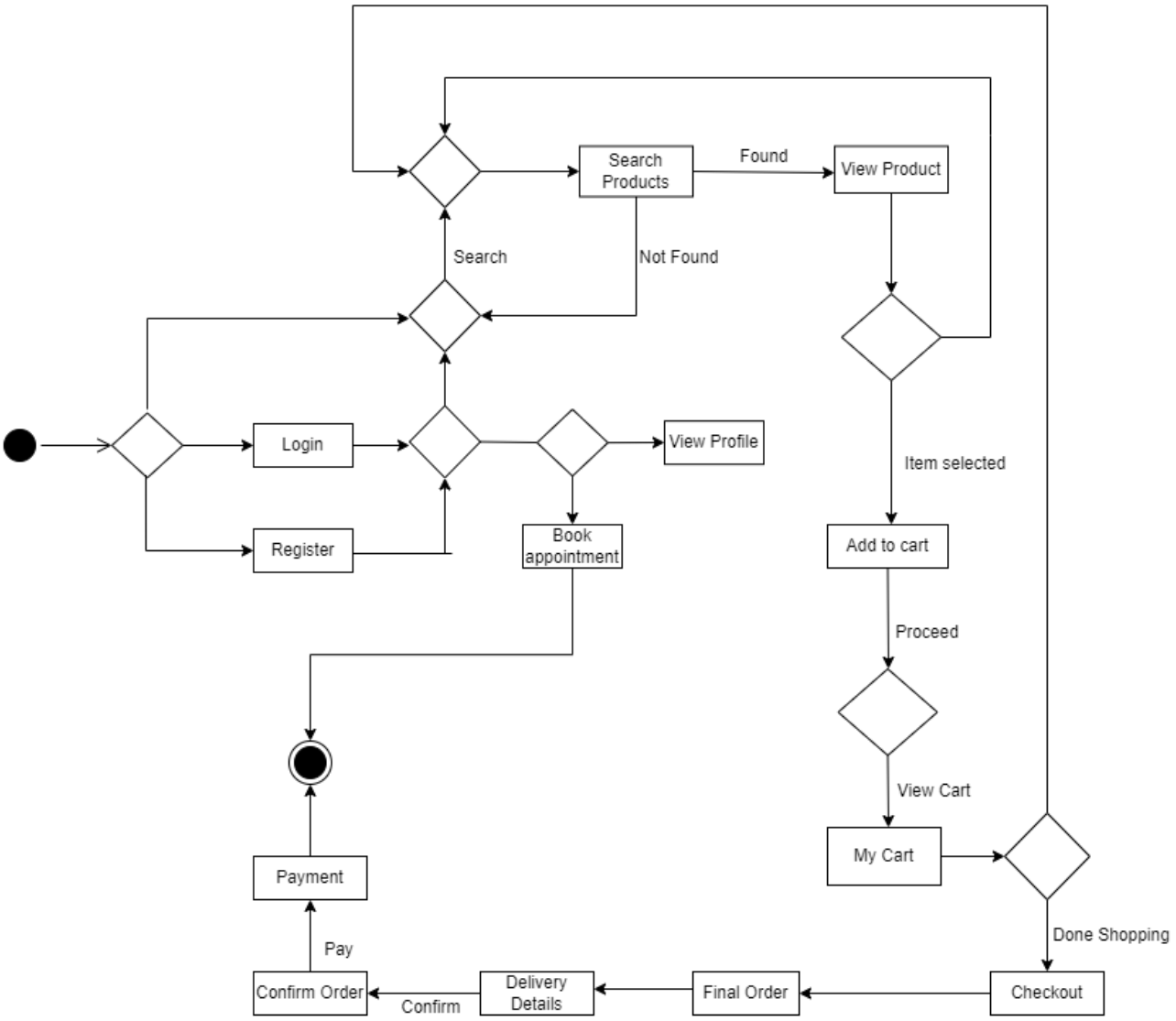


Figure 4.2.3.2: State Chart Diagram for Customer

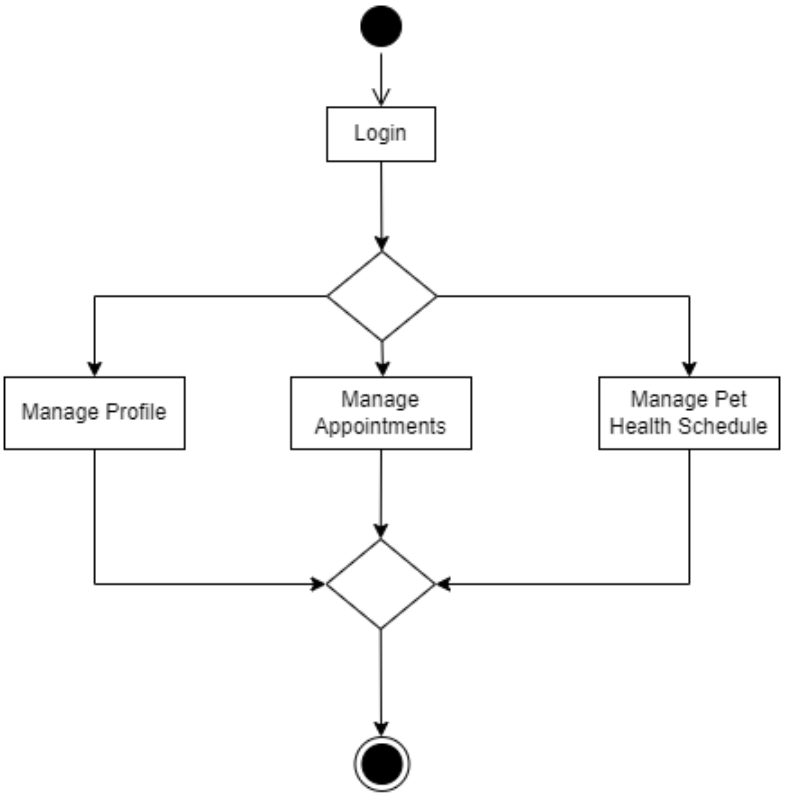


Figure 4.2.3.3: State Chart Diagram for Veterinarian

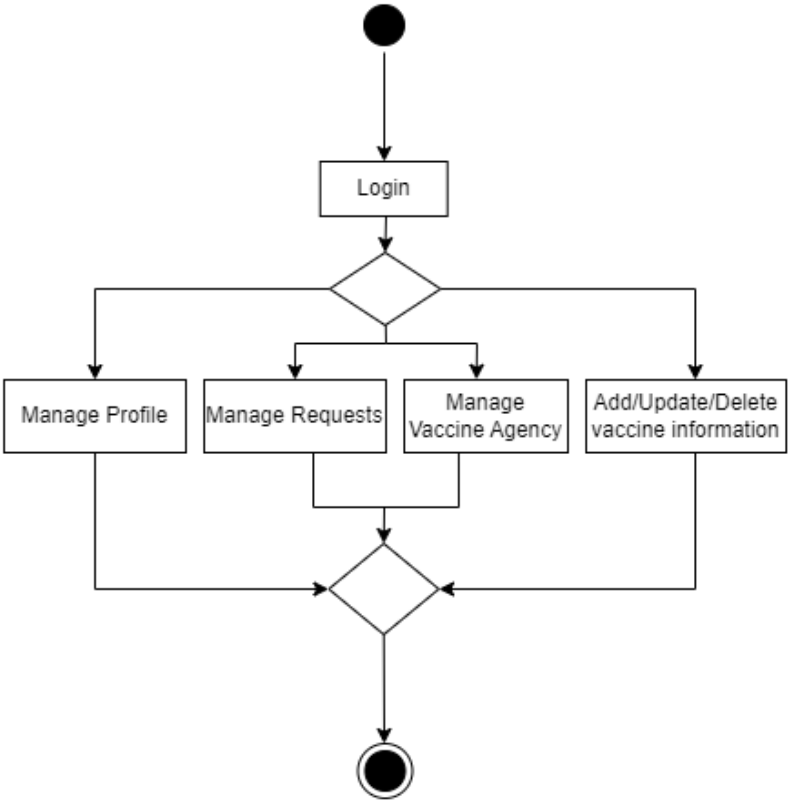


Figure 4.2.3.4: State Chart Diagram for Vaccine Center

4.2.4 Activity Diagram

The dynamic features of the system are described in the activity diagram. An activity diagram is essentially a flowchart that shows how one activity leads to another. The action might be referred to as a system operation. One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. Activity diagrams use many features, such as fork, join, etc., to cope with all types of flow control. Activity diagrams are used to build the executable system utilizing forward and reverse engineering approaches, as well as to visualize the dynamic nature of a system. The message portion is the only item the activity diagram is missing. No message flow from one activity to another is shown. Occasionally, an activity diagram is used in place of a flowchart. The diagrams are not flowcharts, despite their appearance.

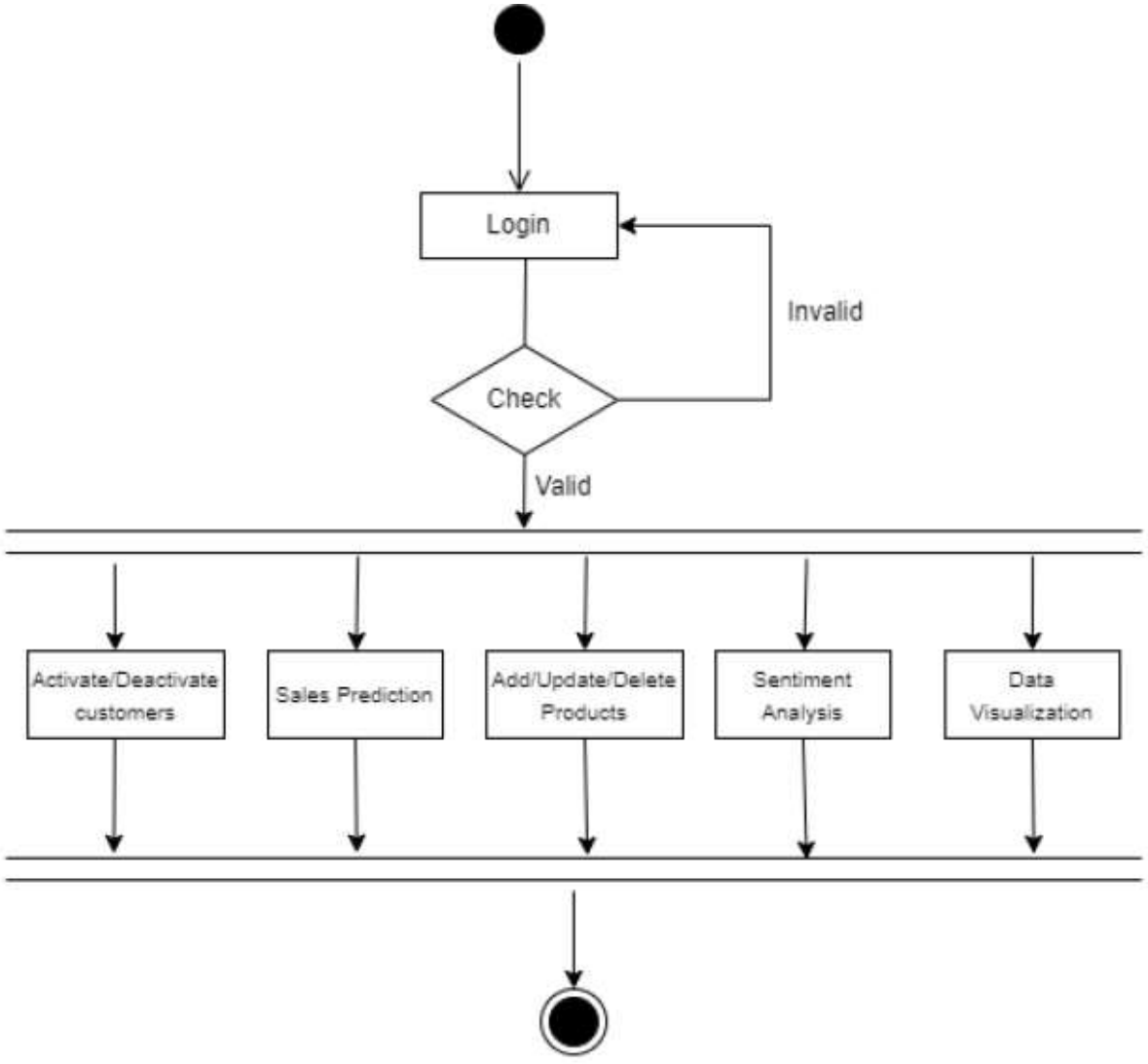


Figure 4.2.4.1 : Activity Diagram for Admin

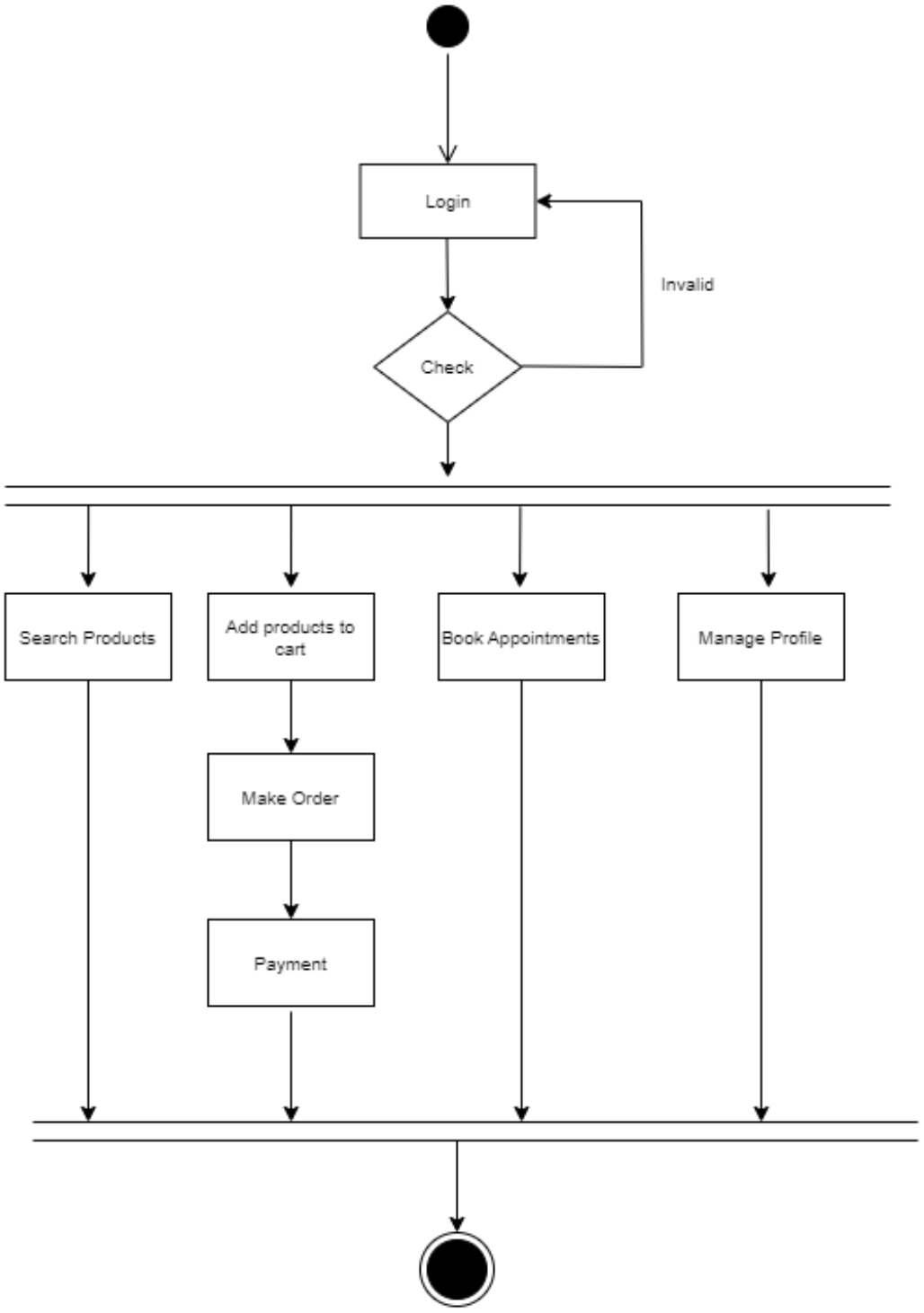


Figure 4.2.4.2: Activity Diagram for Customers

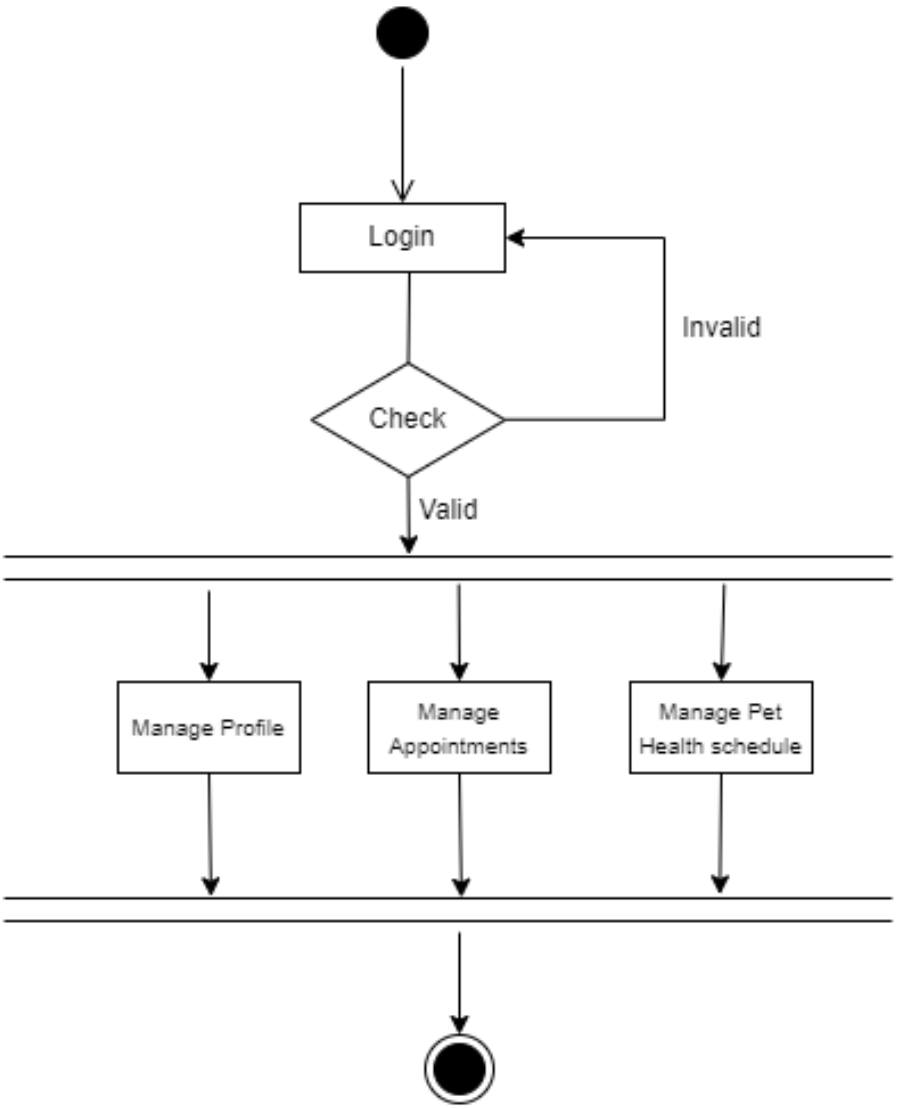


Figure 4.2.4.3: Activity Diagram for Veterinarian

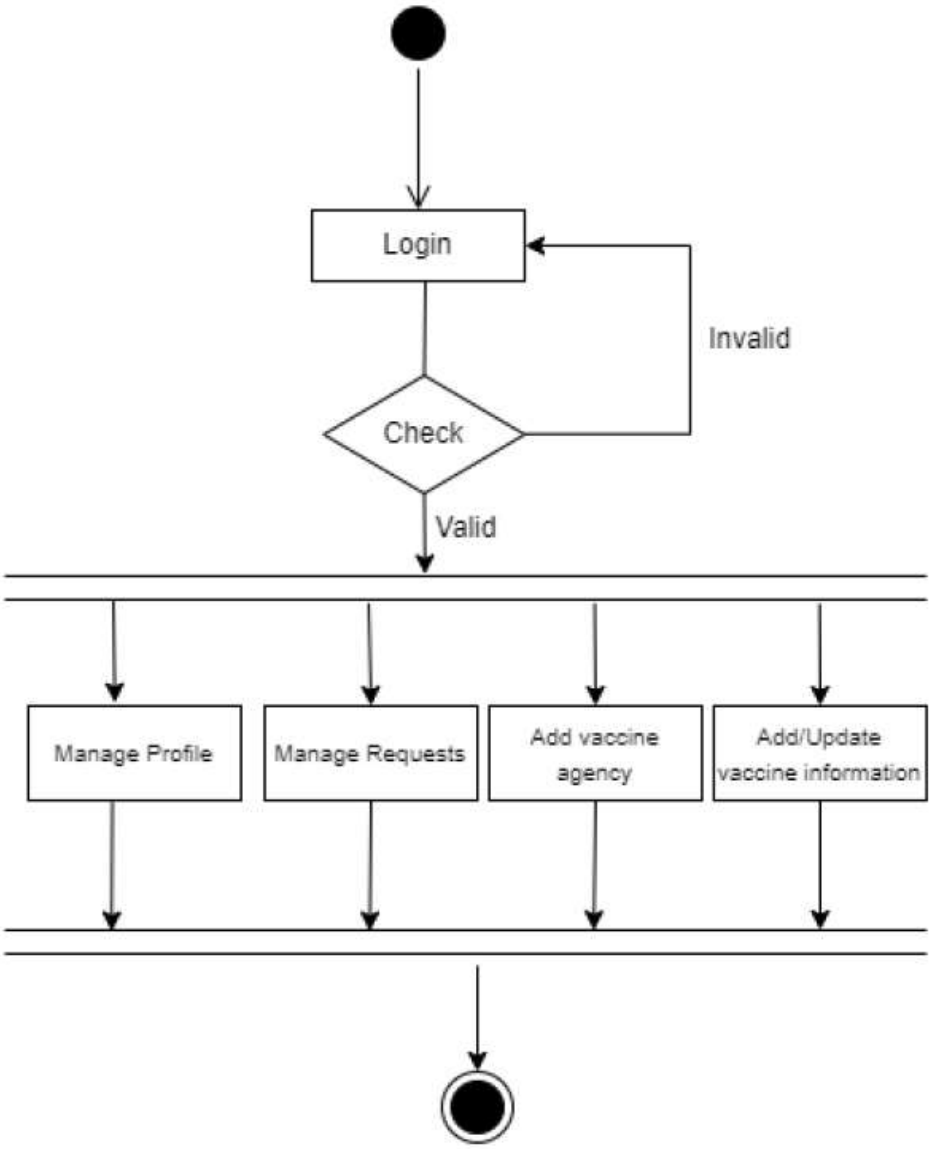


Figure 4.2.4.4: Activity Diagram for Vaccine Center

4.2.5 Class Diagram

Static diagrams include class diagrams. It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualizing, explaining, and documenting various elements of systems. The characteristics and functions of a class are described in a class diagram, along with the restrictions placed on the system. Because they are the only UML diagrams that can be directly mapped with object-oriented languages, class diagrams are frequently employed in the modelling of object-oriented systems. A collection of classes, interfaces, affiliations, collaborations, and constraints are displayed in a class diagram. Another name for it is a structural diagram.

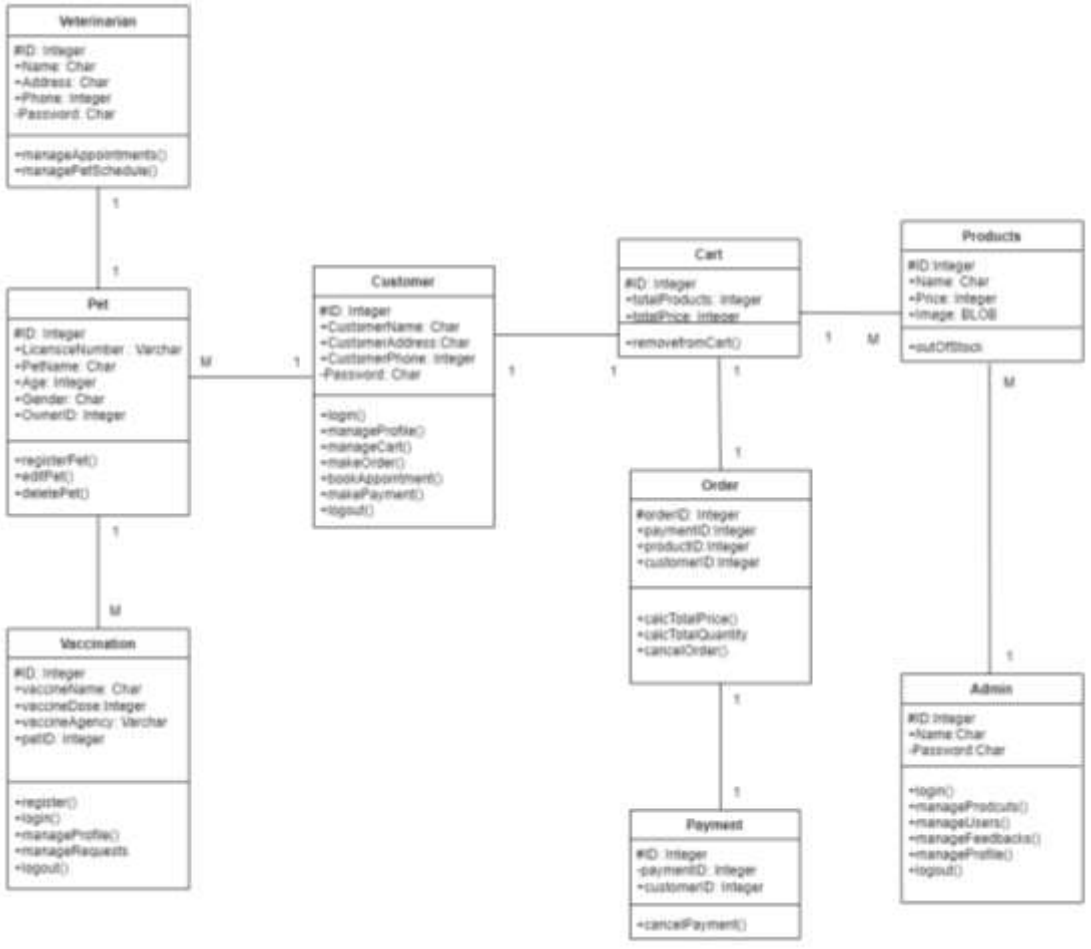


Figure 4.2.5.1 Class diagram for the proposed system

4.2.6 Object Diagram

Since class diagrams are the source of object diagrams, class diagrams are a prerequisite for object diagrams. An instance of a class diagram is represented by an object diagram. Class and object diagrams both use the same fundamental ideas. The static view of a system is also represented by object diagrams, but this static view represents a momentary snapshot of the system. To represent a group of items and their connections as an instance, object diagrams are employed. The purpose of the object diagram can be summarized as:

- Object relationships of a system
- Static view of an interaction.
- Understand object behavior and their relationship from practical perspective.

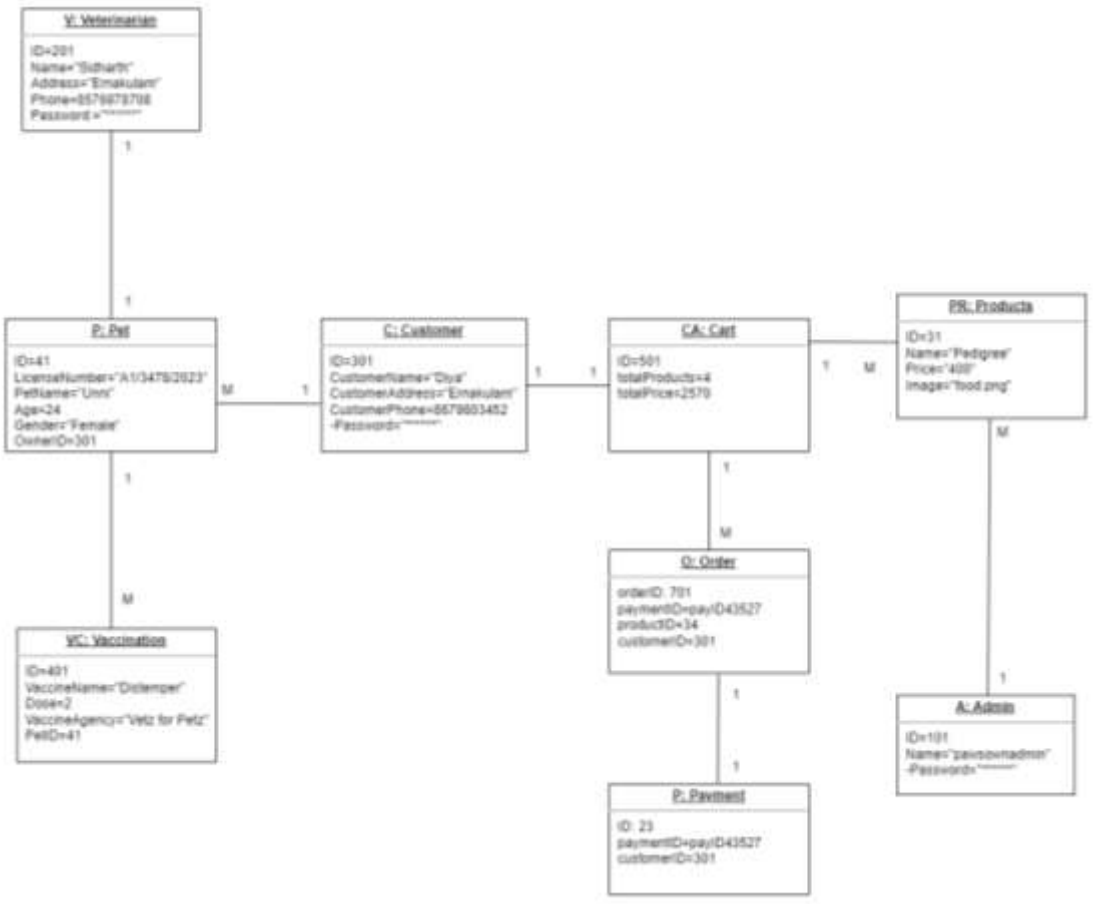


Figure 4.2.6.1: Object diagram for the proposed system

4.2.7 Component Diagram

A specific type of UML diagram is a component diagram. In addition, the goal is distinct from the previous diagrams mentioned. Although it does not describe the system's functionality, it does describe the parts that go into creating that functionality. Therefore, from that perspective, component diagrams are utilized to represent the actual physical parts of a system. These parts include files, libraries, and packages, among others. Another way to think of component diagrams is as a static implementation perspective of a system. Static implementation depicts how the components are arranged at a specific time. A collection of diagrams is used to illustrate the full system because a single component diagram is unable to do so. The purpose of the component diagram can be summarized as:

- Visualize the components of a system.
- Construct executable by using forward and reverse engineering.
- Describe the organization and relationships of the components.

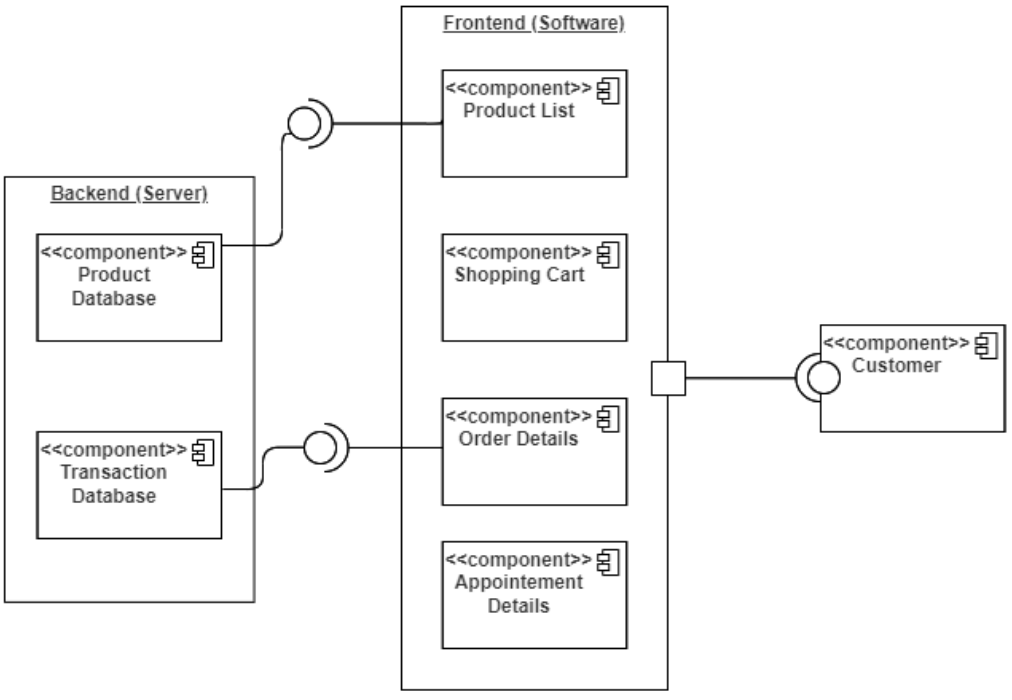


Figure 4.2.7.1: Component diagram for the proposed system

4.2.8 Deployment Diagram

The topology of the physical parts of a system, where the software components are installed, is depicted using deployment diagrams. The static deployment view of a system is described using deployment diagrams. Nodes and their connections are the main components of deployment diagrams. It determines the software deployment strategy on the hardware. It connects the design-created software architecture to the actual system architecture, where the software will run as a node. Communication channels are used to demonstrate the link because there are numerous nodes involved.

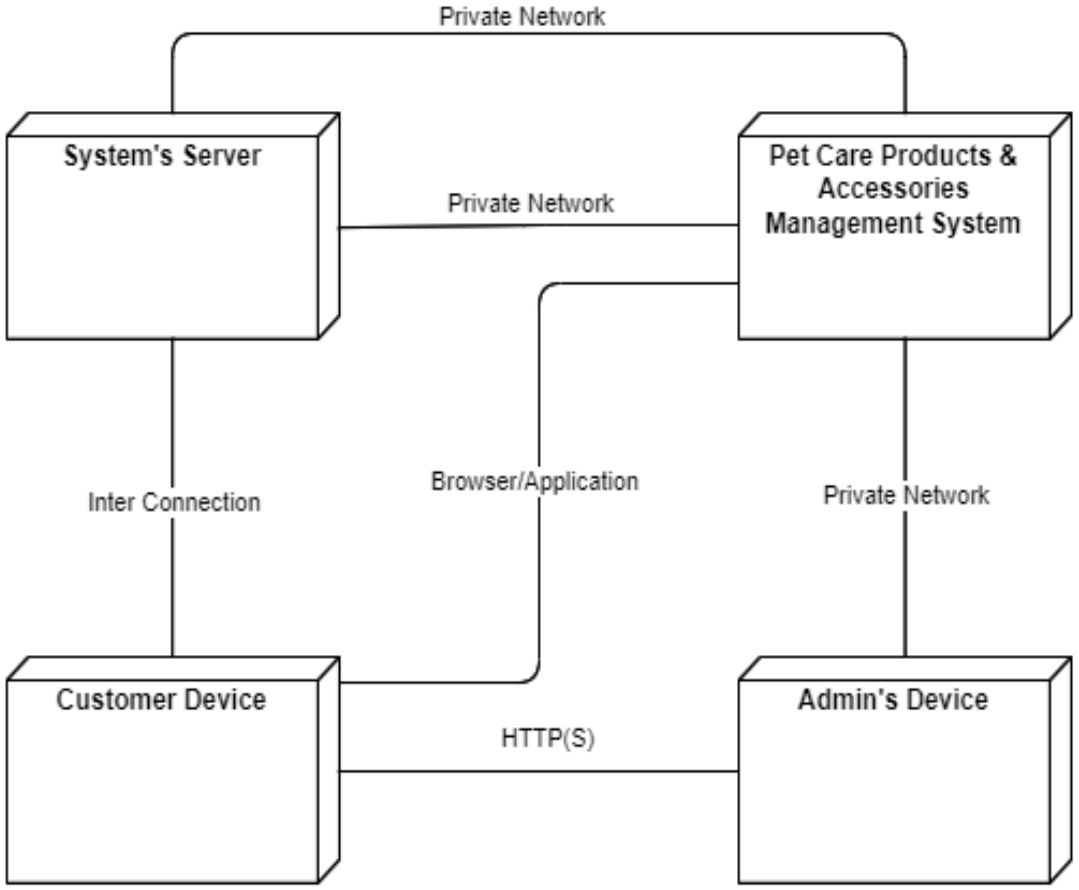


Figure 4.2.8.1: Deployment diagram for the proposed system

4.2.9 Collaboration Diagram

The relationship between the objects in a system is depicted using the cooperation diagram. The same data is shown differently in both the sequence and cooperation diagrams. As it is based on object-oriented programming, it represents the architecture of the object living in the system rather than the flow of messages. An object is made up of various features. The system's various objects are connected to one another. The object's architecture within the system is shown using the collaboration diagram, also referred to as a communication diagram.

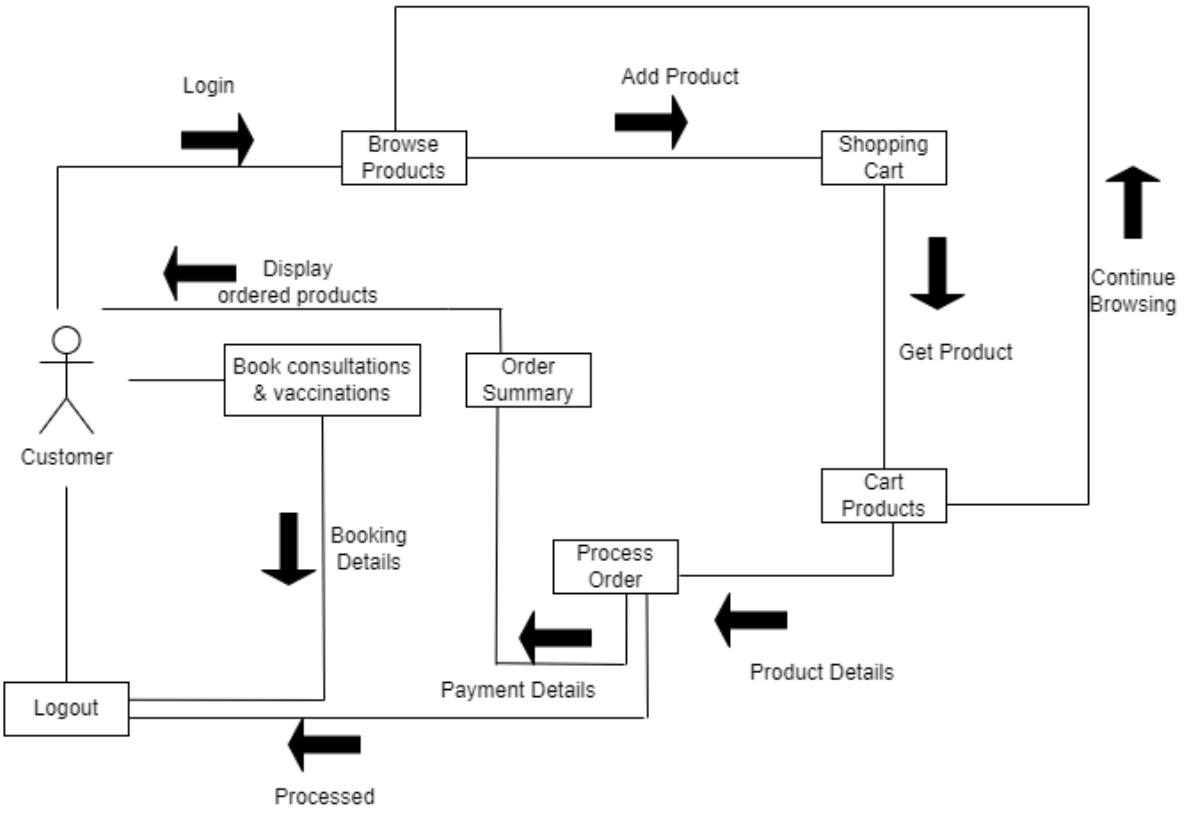


Figure 4.2.9.1: Collaboration diagram for the proposed system

4.3 USER INTERFACE DESIGN USING FIGMA

Fig 4.3.1 Registration Form

Fig 4.3.2 Login Form

Fig 4.3.3 Consultation Form

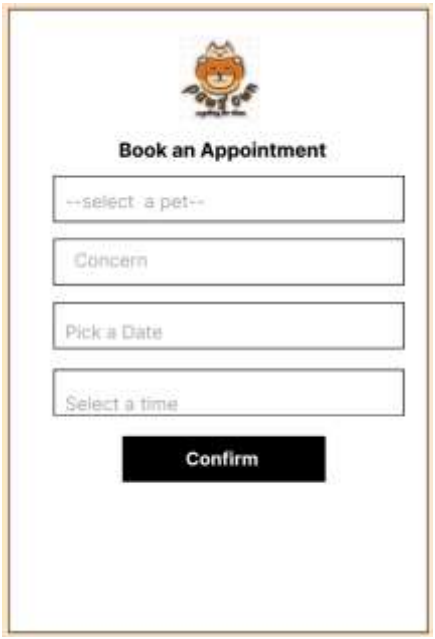


Fig 4.3.4 Vaccination Form

Fig 4.3.5 Customer Home Page

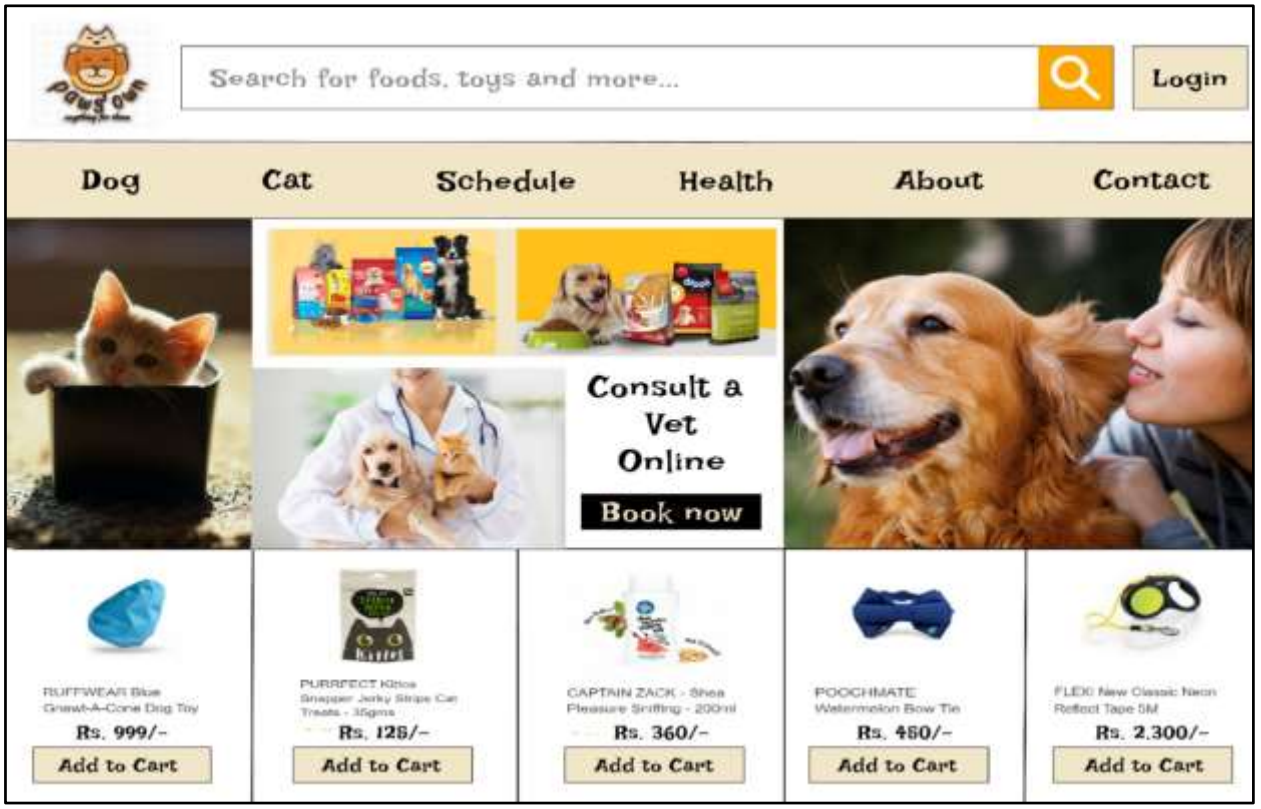
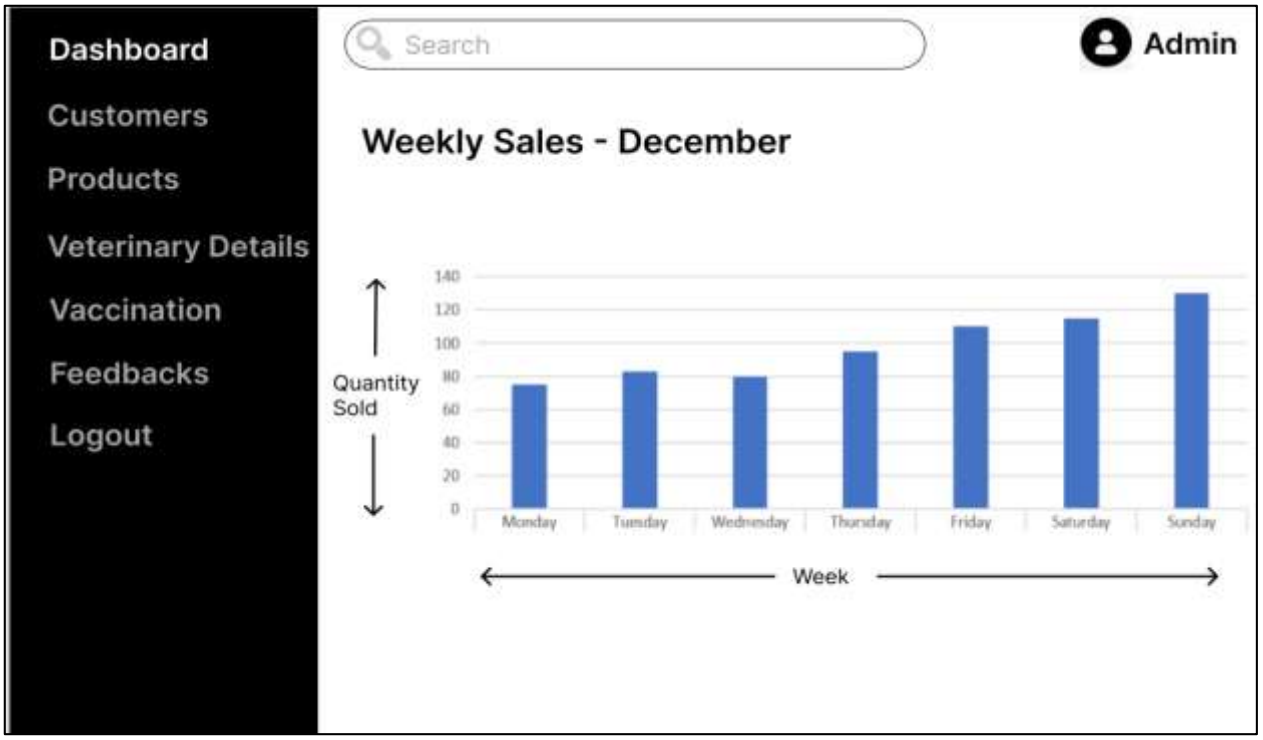


Fig 4.3.6 Sales Prediction



4.4 DATABASE DESIGN

A database is a structured system with the capacity to store information and allows users to retrieve stored information quickly and effectively. Any database's primary goal is its data, which demands protection. There are two stages to the database design process. The user needs are obtained in the first step, and a database is created to as clearly as possible meet these criteria. This process, known as information level design, is carried out independently of all DBMSs. The design for the specific DBMS that will be used to construct the system in issue is converted from an information level design to a design in the second stage. Physical Level Design is the stage where the characteristics of the particular DBMS that will be used are discussed. Parallel to the system design is a database design. Data Integrity and Data Independence are goals of the database's data organization.

4.4.1 Relational Database Management System (RDBMS)

A relational model shows the database as a group of relationships. A table of values or a file of records is how each relation looks. In the formal language of the relational model, a row is referred to as a tuple, a column heading is referred to as an attribute, and the table is referred to as a relation. The tables in a relational database are organized into groups and given individual names. In a story, a row corresponds to a group of connected values.

4.4.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. The formal method of normalizing data structures in a way that reduces duplication and fosters integrity. Using the normalization process, unnecessary fields are removed and a huge table is divided into several smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Keys and relationships are two notions used in the standard form of data modelling. A row in a table is uniquely identified by a key. Primary keys and foreign keys are the two different kinds of keys. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalized. It means arranging things in their regular order. By using normalization, the application developer aims to establish a sensible arrangement of the data into appropriate tables and columns, where names may be quickly related to the data by the user. By removing repetitive groups from data, normalization prevents data redundancy, which places a heavy demand on the computer's resources.

4.4.3 Sanitization

Sanitizing data entails deleting any elements that are prohibited. One of the most frequent duties in a web application is sanitizing user input. PHP offers a native filter extension that you can use to sanitize the data, such as email addresses, URLs, IP addresses, etc., to make the process easier. The extension for the PHP filter: Sanities and validate external input using PHP filters. Many of the functions required for validating user input are included in the PHP filter extension, which is also made to make the process of sanitizing data simpler and faster. Using the flag in the example, this method ensures that all characters other than letters, numerals, and the following characters are removed from the code #$%&'*+-=?_`{|}~@.[] . When the input format is flexible but yet relatively predictable, such as with phone numbers or other free-text fields, sanitizing inputs can be a useful alternative. You can sanitize inputs in a number different methods, such as by using a whitelist, a blacklist, or escape input.

4.4.4 Indexing

Indexing is used to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed. The index is a type of data structure. It is used to locate and access the data in a database table quickly. Indexes can be created using some database column.

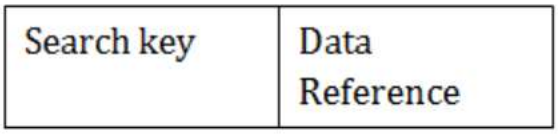


Figure 4.4.3.1: Structure of Index

The primary key or candidate key of the table is duplicated in the first column, which is the Search key. To make it easier to find the related data, these values are kept in sorted order. The Data Reference or Pointer column, which appears in the second row, comprises a set of pointers that each represent the location of the disc block where the specific key value is stored. The indexing properties of a database are used to define its indexing. Primary indexing and secondary indexing are the two primary categories of indexing techniques. The Primary Index is a fixed-length, ordered file containing two fields. A main key is the first field, and the second field points to that particular data block. The entries in the index table are always related to one another in the primary index.

The secondary Index in DBMS can be generated by a field which has a unique value for each record, and it should be a candidate key. It is also known as a non-clustering index. This two-level database indexing technique is used to reduce the mapping size of the first level. For the first level, a large range of numbers is selected because of this; the mapping size always remains small.

4.5 TABLE DESIGN

1. tbl_userdetails

Primary Key: **userID**

No:	Fieldname	Data type	Key Constraints	Description
1	userID	int (3)	Primary Key	Primary Key of tbl_userdetails
2	userName	varchar(20)	NOT NULL	User Name
3	userEmail	varchar (20)	NOT NULL	User Email ID
4	userPhone	varchar (20)	NOT NULL	Phone number of user
5	userRole	varchar (20)	NOT NULL	Role of user
6	userPassword	varchar(20)	NOT NULL	User Password
7	userStatus	tinyint(1)	NOT NULL	Status of user
8	userHousename	varchar(40)	NULL	House Name of user
9	userPostOffice	varchar(40)	NULL	Post Office of user
10	userLocality	varchar(20)	NULL	Locality of user
11	userDistrict	varchar(20)	NULL	District of user
12	userTown	varchar (20)	NULL	Town of user
13	userPincode	int(6)	NULL	Pincode of user

2. tbl_productdetails

Primary Key: **productID**

No:	Fieldname	Data type	Key Constraints	Description
1	productID	Int(3)	Primary Key	Primary Key of tbl_productdetails
2	productFor	varchar(20)	NOT NULL	Product is for Dog/Cat
3	productCategory	varchar (20)	NOT NULL	Category of Food
4	productName	varchar (20)	NOT NULL	Name of the product
5	productSubcategory	varchar (20)	NOT NULL	Sub-category of product
6	productSubname	varchar(20)	NOT NULL	Subname of product
7	productPrice	varchar (4)	NOT NULL	Price of the product
8	productDescription	varchar(20)	NOT NULL	Description of the products
9	productImage	longblob	NOT NULL	Image of product
10	productStatus	tinyint(1)	NOT NULL	Status of the user

3. tbl_stock

Primary Key : **stockID**

Foreign Key : **productID** references table **tbl_productdetails**

No:	Fieldname	Data type	Key Constraints	Description
1	stockID	int (3)	Primary Key	Primary Key of tbl_stock
2	productID	int(3)	NOT NULL	Foreign key from tbl_productdetails
3	productWeight	int(3)	NOT NULL	Weight of product
4	productPrice	varchar(20)	NOT NULL	Price of product
5	productStock	varchar(20)	NOT NULL	Stock of products
6	stockStatus	varchar(20)	NOT NULL	Status of tbl_stock

4. tbl_petdetails

Primary Key : **petID**

Foreign Key : **customerID** references table **tbl_userdetails**

No:	Fieldname	Data type	Key Constraints	Description
1	petID	int (3)	Primary Key	Primary Key of tbl_petdetails
2	customerID	int (3)	NOT NULL	Foreign key from tbl_userdetails
3	petLicensenumber	varchar (20)	NOT NULL	License Number of Pet
4	petName	varchar (20)	NOT NULL	Name of Pet
5	petSpecies	varchar(20)	NOT NULL	Species of Pet
6	petBreed	varchar(20)	NOT NULL	Breed of Pet
7	petAge	varchar(20)	NOT NULL	Pet age
8	petGender	varchar(1)	NOT NULL	Gender of Pet
9	petColor	varchar(40)	NOT NULL	Color of Pet
10	petStatus	tinyint(1)	NOT NULL	Status of Pet

5. tbl_petshedule

Primary Key : **scheduleID**

No:	Fieldname	Data type	Key Constraints	Description
1	scheduleID	Int (3)	Primary Key	Primary Key of tbl_petschedule
2	scheduleTitle	varchar (30)	NOT NULL	Title of the schedule
3	scheduleDescription	varchar (300)	NOT NULL	Description about the schedule
4	scheduleStatus	tinyint (1)	NOT NULL	Status of the schedule

6. tbl_appointmentdetails

Primary Key : **appointmentID**

Foreign Key 1 : **customerID** references table **tbl_userdetails**

Foreign Key 2 : **petID** references table **tbl_petdetails**

No:	Fieldname	Data type	Key Constraints	Description
1	appointmentID	int(3)	Primary Key	Primary Key of tbl_appointmentdetails
2	customerID	int (3)	Foreign Key	Foreign key from tbl_userdeatils
3	petID	int(3)	Foreign Key	Foreign key from tbl_petdetails
4	concernAboutPet	varchar(30)	NOT NULL	Concern about the pet
5	consultationDate	date	NOT NULL	Date of consultation
6	consultationTime	time	NOT NULL	Tome of consultation
7	appointmentMedicine	varchar(80)	NULL	Suggested medicines
8	appointmentPrescription	varchar(80)	NULL	Prescription by Vet
9	appointmentStatus	tinyint(1)	NOT NULL	Status of appointment

7. tbl_vaccineagency

Primary Key : **vaccineagencyID**

No:	Fieldname	Data type	Key Constraints	Description
1	vaccineagencyID	int (3)	Primary Key	Primary Key of tbl_vaccinagency
2	agencyName	varchar(30)	NOT NULL	Name of vaccine agency
3	agencyEmail	varchar(20)	NOT NULL	Email of vaccine agency
4	agencyPhone	varchar(10)	NOT NULL	Contact number of agency
5	agencyLocattion	varchar(100)	NOT NULL	Location of Agency
6	agencyStatus	tinyint()	NOT NULL	Status of vaccine agency

8. tbl_vaccinedetails

Primary Key : **vaccineID**

Foreign Key : **vaccineagencyID** references table **tbl_vaccineagency**

No:	Fieldname	Data type	Key Constraints	Description
1	vaccineID	int (3)	Primary Key	Primary Key of tbl_vaccinedetails
2	vaccineagencyID	int(3)	NOT NULL	Foreign key from tbl_vaccineagency
3	vaccineID	int(3)	NOT NULL	Foreign key from tbl_vaccinedetails
4	vaccineDose	varchar(100)	NOT NULL	Dose per visit
5	boosterRecommend	varchar(100)	NOT NULL	Booster Gap
6	vaccineComments	Varchar(300)	NOT NULL	Vaccine Description
7	vaccineAvailability	int(3)	NOT NULL	Vaccine available or not
8	vaccineStatus	tinyint(3)	NOT NULL	Status of vaccine

9. tbl_vaccination

Primary Key : **vaccineID**

Foreign Key 1 : **customerID** references table **tbl_userdetails**

Foreign Key 2 : **petID** references table **tbl_petdetails**

Foreign Key 3 : **vaccineagencyID** references table **tbl_vaccineagency**

No:	Fieldname	Data type	Key Constraints	Description
1	vaccineID	int (3)	Primary Key	Primary Key of tbl_bookvaccine
2	vaccineagencyID	int(3)	Foreign Key	Foreign Key from tbl_vaccineagency
3	customarID	int(3)	Foreign Key	Foreign key from tbl_userdetails
4	petID	int(3)	Foreign Key	Foreign key from tbl_petdetails
5	customerTown	varchar(50)	NOT NULL	Customer Town
6	customerDistrict	date	NOT NULL	Custpmer District
7	vaccineName	varchar(30)	NOT NULL	Name of vaccine to be givan
8	vaccinationTime	time	NOT NULL	Vaccination Time
9	vaccinatedDate	date	NULL	Date on which vaccine is given
10	vaccineStatus	tinyint(1)	NOT NULL	Status of vaccine

10. tbl_cart

Primary Key: **cartID**

Foreign Key 1: **customerID** references table **tbl_userdetails**

Foreign Key 2: **productID** references table **tbl_productdetails**

Foreign Key 3: **stockID** references table **tbl_stock**

No:	Fieldname	Data type	Key Constraints	Description
1	cartID	int (3)	Primary Key	Primary Key of tbl_cart
2	customerID	int (3)	Foreign Key	Foreign key from tbl_userdetails
3	productID	int(3)	Foreign Key	Foreign key from tbl_productdetails
4	stockID	int(3)	Foreign Key	Foreign key from tbl_stock
5	productCount	int(2)	NOT NULL	Quantity of product
6	cartStatus	tinyint (1)	NOT NULL	Status of cart

11. tbl_payment

Primary Key : **transactionID**

Foreign Key: **customerID** references table **tbl_userdetails**

No:	Fieldname	Data type	Key Constraints	Description
1	transactionID	int (3)	Primary Key	Primary Key of tbl_payment
2	paymentID	varchar (40)	NOT NULL	Payment ID of tbl_payment
3	razorpayOrderID	varchar (40)	NOT NULL	Order ID of Razor pay
4	customerID	int (3)	Foreign Key	Foreign Key from User Details table
5	paymentStatus	tinyint (1)	NOT NULL	Status of payment

12. tbl_order

Primary Key: **orderID**

Foreign Key 1**: customerID** references table **tbl_userdetails**

Foreign Key 2: **transID** references table **tbl_payment**

No:	Fieldname	Data type	Key Constraints	Description
1	orderID	int (3)	Primary Key	Primary Key of tbl_order
2	customerID	int (3)	Foreign Key	Foreign key from tbl_userdetails
3	transID	int (3)	Foreign Key	Foreign key from tbl_payment
4	orderDate	date	NOT NULL	Date of order
5	orderTime	time	NOT NULL	Time of order
6	shippedDate	date	NULL	shippedDate
7	deliveredDate	date	NULL	Delivered Date
8	orderStatus	tinyint (1)	NOT NULL	Status of order

13. tbl_orderitems

Primary Key: **orderitemsID**

Foreign Key 1: **orderID** references table **tbl_order**

Foreign Key 2: **productID** references table **tbl_productdetails**

Foreign Key 3**: stockID** references table **tbl_stock**

No :	Fieldname	Data type	Key Constraints	Description
1	orderitemsID	int (3)	Primary Key	Primary Key of tbl_orderitems
2	orderID	int (3)	Foreign Key	Foreign key from tbl_order
3	productID	int (3)	Foreign Key	Foreign key from tbl_productdetails
4	stockID	int(3)	Foreign Key	Foreign key from tbl_stock
4	productPrice	varchar (4)	NOT NULL	Price of product
5	productQuantity	int(25)	NOT NULL	Total quantity of product
6	orderitemsStatus	tinyint (1)	NOT NULL	Status of Order Items

14. tbl_feedback

Primary Key: **feedbackID**

No:	Fieldname	Data type	Key Constraints	Description
1	feedbackID	int (3)	Primary Key	Primary Key of tbl_feedback
2	customerID	int(3)	NOT NULL	Foreign Key from tbl_userdetails
3	customerFeedback	varchar (20)	NOT NULL	Feedback from customer
5	feedbackStatus	tinyint (1)	NOT NULL	Status of feedback

CHAPTER 5

SYTEM TESTING

5.1 INTRODUCTION

Software testing is the process of assessing and confirming that a software application or product performs as intended. Testing has advantages such as bug prevention, lower development costs, and better performance. The terms "software testing" and "verification and validation" are frequently used together. Validation is the process of examining or evaluating a product, including software, to determine whether it complies with all relevant specifications. One type of verification, software testing, uses methods including reviews, analyses, inspections, and walkthroughs as well. Verifying that what has been specified matches what the user truly desired is the process of validation. The processes of static analysis and dynamic analysis are additional ones that are frequently related to software testing. Static analysis examines the software's source code, searching for issues and obtaining statistics without actually running the code. Dynamic analysis examines how software behaves while it is running in order to offer data like execution traces, timing profiles, and test coverage details. Testing is a collection of activities that can be planned ahead of time and carried out in a methodical manner. Testing starts with individual modules and progresses all the way to system integration for computer-based systems. There are many rules that can be used as testing objectives, and testing is necessary for the system testing objectives to be successful. They are:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a test is successfully carried out in accordance with the aforementioned aims, it will reveal software bugs. Additionally, testing shows that the software functions seem to operate in accordance with the specification and that the performance requirements seem to have been satisfied. Programs can be tested in three different ways: for correctness, implementation effectiveness, and computational power. Testing for correctness is meant to ensure that a programs performs exactly as it was intended to. This is much harder than it might initially seem, especially for big programs.

5.1 TEST PLAN

A test plan suggests a number of required steps that need be taken in order to complete various testing methodologies. The activity that is to be taken is outlined in the test plan. A computer programme, its documentation, and associated data structures are all created by software

developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure it fulfils the purpose for which it was intended. In order to solve the inherent issues with allowing the builder evaluate what they have developed, there is an independent test group (ITG). Testing's precise goals should be laid forth in quantifiable language. The test strategy should include information on the mean time to failure, cost to detect and correct problems, remaining defect density or frequency of occurrence, and test work hours per regression test The levels of testing include:

- Unit testing
- Integration Testing
- Validation Testing or System Testing
- Output Testing or User Acceptance Testing
- Automation Testing
- Selenium Testing

5.2.1 Unit Testing

A software testing technique known as unit testing involves testing individual software units, such as groups of computer programme modules, usage procedures, and operating procedures, to see if they are acceptable for use. It is a testing technique whereby the developer himself tests each individual module to ascertain whether there is a problem. It has a relationship on how well the individual modules perform. Unit testing is a sort of software testing in which individual software components are tested. During the creation of an application, unit testing of the programme is done. An individual component could be a technique or a specific function. The developer typically conducts unit testing. Unit testing is the initial level of testing performed prior to integration testing in the SDLC or V Model. One such testing method is unit testing, which is typically carried out by developers.

5.2.2 Integration Testing

Software components are logically connected and tested as a unit in a type of testing called integration testing. Multiple software modules created by various programmers make up a typical software project. This level of testing aims to identify issues with how various software modules interact when they are combined. The programme as a whole is tested. Since it is difficult to pinpoint the causes due to the magnitude of the entire programme, correction is difficult. As soon as these errors are corrected, new ones appear, and the process keeps repeating itself. Additionally,

many programme structure variations were eliminated, and a unique programme structure was created.

5.2.3 Validation Testing or System Testing

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfils its intended use when deployed on appropriate environment. It answers to the question, Are we building the right product? The definition of validation testing in software engineering is in place to determine if the existing system complies with the system requirements and performs the dedicated functions for which it is designed along with meeting the goals and needs of the organization. This mode of testing is extremely important especially if you want to be one of the best software testers. The software verification and validation testing is the process after the validation testing stage is secondary to verification testing.

5.2.4 Output Testing or User Acceptance Testing

User acceptance testing (UAT), also known as application testing or end-user testing, is a stage of the software development process when the target user group tests the product in the real world. Before the tested product is made available to its target market, UAT is frequently the final stage of the software testing process. UAT seeks to confirm that software can handle practical activities and execute in accordance with development requirements. Prior to its official release, users are given the chance to interact with the software in UAT to determine whether any features have been missed or if it has any issues. UAT can be carried out internally with volunteers, utilising paid test subjects to utilise the programme, or by allowing users to download the test version as a free trial. Early testers provide feedback to the creators, who then make any necessary adjustments before making the software available for purchase. UAT increases user transparency while being an efficient way to guarantee quality in terms of both time and software costs. Additionally, UAT gives developers access to actual instances and data, and if it goes well, it can confirm business requirements.

5.2.5 Automation Testing

Automation Testing is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a

human sitting in front of a computer carefully executing the test steps. The automation testing software can also enter test data into the System under Test, compare expected and actual results and generate detailed test reports. Software Test Automation demands considerable investments of money and resources. Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual testing altogether.

5.2.6 Selenium Testing

All web application developers should be well-versed in Selenium, an open-source, automated testing tool. Selenium automated testing is the term used to describe testing carried out with Selenium. Selenium, however, is a group of tools that each serve a specific purpose for Selenium automated testing. The Selenium toolbox is used by Selenium automation to execute tests on many browser instances. You may test simultaneously across numerous instances and machines with Selenium Grid. System testing, End-to-end testing, Compatibility testing, Regression testing, Integration testing, and Performance testing are just a few of the many types of testing that can be done with it. Selenium is not just one tool; rather, it is a group of programmes that each address a different organization's testing requirements. It has four components:

- **Selenium Integrated Development Environment (IDE):** Integrated Development Environment, or IDE for short, is a Firefox browser plug-in. Since this is the most basic framework, developers must migrate to Selenium RC for complex test scenarios.
- **Selenium Remote Control (RC):** Remote Control, or RC, enables programmers to write code in the language of their choice. Numerous programming languages, including Java, C#, Python, etc., are supported by Selenium RC.
- **Selenium Web Driver:** The actions made by a web browser are automated and managed by Web Driver. It often interacts with the browser rather than relying on JavaScript to govern its actions. RC and Web Driver both support other programming languages like Java, C#, Python, Ruby, et
- **Selenium Grid:** Grid is used in conjunction with RC to run tests simultaneously across several browsers.

Test Case 1- Registration

Code

```
package spl;
import org.openqa.selenium.By;
public class selproRegister
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver","C:\\Users\\user\\Downloads\\chromedriver_win32 (3)\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.get("http://localhost/minlproject/register.php");
        driver.manage().window().maximize();
        driver.findElement(By.id("name")).sendKeys("Geya Shibu");
        driver.findElement(By.id("email")).sendKeys("geyamerinshibu@mca.ajce.in");
        driver.findElement(By.id("phone")).sendKeys("9788789006");
        driver.findElement(By.id("password")).sendKeys("123456");
        driver.findElement(By.id("cpassword")).sendKeys("123456");
        driver.findElement(By.id("submitbutton")).click();
        String at=driver.getTitle();
        String et="Register";
        driver.close();
        if(at.equalsIgnoreCase(et))
        {
            System.out.println("Test Passed");
        }
        else
        {
            System.out.println("Test Failed");
        }
    }
}
```

Output Screenshot

```
Starting ChromeDriver 111.0.5563.64 (c710e93d5b63b7095afe8c2c17df34408078439d-refs/branch-heads/5563@{#995}) on port 43528
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Apr 01, 2023 2:52:09 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Test Passed
```

Test report

Test Case 1

Project Name: PAWS' OWN					
Registration Test Case					
Test Case ID: Test 1			**Test Designed By: Lakshmi Sunil**		
Test Priority(Low/Medium/High): High			**Test Designed Date: 01/04/2023**		
Module Name: Registration			**Test Executed By : DR. BIJIMOL T. K.**		
Test Title : Register new user with valid Name, Email, Phone and Password			**Test Execution Date:01/04/2023**		
Description: Registration Page					
Pre-Condition : New inputs must be valid					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**

1	Navigate to Registration Page		Registration Page	Navigated to Registration Page	Pass
2	Enter valid Name	Name: Geya Merin Shibu	No error message	New account created	Pass
3	Enter valid email	Email: geyamerinshibu@gmail.com			
4	Enter valid phone number	Phone:8763901111			
5	Enter valid password	password:12345678			
6	Re-enter password	Password:12345678			
7	Click on Login button				
8	Enter invalid Name or NULL	Name:9Sreeja Name: NULL	Error message should be displayed	New account registration failed	Pass
9	Enter invalid Email or null	Email:@rty.l@gmail.com Email: NULL			
10	Enter invalid Phone or NULL	Phone:785 Phone:NULL			
12	Enter invalid password or NULL	Password:abc Password:NULL			
13	Click on login button				
Post-Condition: All inputs are validated and account is created successfully by saving the entries into the database.					

Test Case 2: Login

Code

```
package spl;
import org.openqa.selenium.By;
public class selprol
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver","C:\\Users\\user\\Downloads\\chromedriver_win32 (3)\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();

        driver.get("http://localhost/miniproject/login.php");
        driver.manage().window().maximize();
        driver.findElement(By.id("em")).sendKeys("lakshmisunil@mca.ajce.in");
        driver.findElement(By.id("password")).sendKeys("123456");
        driver.findElement(By.id("submit")).click();
        String at=driver.getTitle();
        String et="Login";
        driver.close();
        if(at.equalsIgnoreCase(et))
        {
            System.out.println("Test Passed");
        }
        else
        {
            System.out.println("Test Failed");
        }
    }
}
```

Screenshot

```
Starting ChromeDriver 111.0.5563.64 (c710e93d5b63b7095afe8c2c17df34408078439d-refs/branch-heads/5563@{#995}) on port 43528
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Apr 01, 2023 2:52:09 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Test Passed
```

Test report

Test Case 2

Project Name: PAWS' OWN					
Login Test Case					
Test Case ID: Test 2			**Test Designed By: Lakshmi Sunil**		
Test Priority(Low/Medium/High): High			**Test Designed Date: 01/04/2023**		
Module Name: Login			**Test Executed By : DR. BIJIMOL T. K.**		
Test Title : Verify Login with username and password			**Test Execution Date:01/04/2023**		
Description: Test Login Page					
Pre-Condition : User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to Login Page		Login Page	Navigated to Login Page	Pass

2	Enter valid username	username: lakshmisunil@mca.ajce.in	User should be able to Login	User logged in	Pass
3	Enter valid password	password: 1234567			
4	Click on Login button				
5	Enter invalid username or null	username: lakshmi@gmail.com username: NULL	User should not be able to Login	User is not logged in an error message displayed	Pass
6	Enter invalid password or null	password: 1234 password: NULL			
7	Click on login button				
Post-Condition: User is validated with database and successfully logged in to the system					

Test Case 3: Submit Feedback

Code

```
package spl;
import org.openqa.selenium.By;
public class selproFeedback
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver","C:\\Users\\user\\Downloads\\chromedriver_win32 (3)\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.get("http://localhost/miniproject/login.php");
        driver.manage().window().maximize();
        driver.findElement(By.id("em")).sendKeys("lakshmisunil@mca.ajce.in");
        driver.findElement(By.id("password")).sendKeys("123456");
        driver.findElement(By.id("submit")).click();
        driver.get("http://localhost/miniproject/contact.php");
        driver.manage().window().maximize();
        driver.findElement(By.id("cusfeedback")).sendKeys("Good");
        driver.findElement(By.id("submitbutton")).click();
        String at=driver.getTitle();
        String et="Feedback Form";
        driver.close();
        if(at.equalsIgnoreCase(et))
        {
            System.out.println("Test Failed");
        }
        else
        {
            System.out.println("Test Passed");
        }
    }
}
```

Screenshot

Test Report

Test Case 3

Project Name: PAWS' OWN					
Feedback Form Test Case					
Test Case ID: Test_3			**Test Designed By: LAKSHMI SUNIL**		
Test Priority(Low/Medium/High): High			**Test Designed Date: 01/04/2023**		
Module Name: Feedback Form			**Test Executed By : DR. BIJIMOL T. K.**		
Test Title : Verify Feedback (alphabets only)			**Test Execution Date:01/04/2023**		
Description: Test Feedback Form					
Pre-Condition : Feedback must contain only alphabets					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to Feedback Page		Feedback Page	Navigated to Feedback Page	Pass
2	Enter valid feedback	Feedback : Excellent products and services	Feedback submitted successfully	Feedback submitted	Pass
4	Click on Login button				
5	Enter feedbacks with numbers	Feedback: 8#hufufg	Feedback cannot be submitted	Feedback is not submitted	Pass
7	Click on login button				
Post-Condition: Valid feedback is submitted successfully					

Test Case 4: Pet Registration Form

Code

```
package spl;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class selproRegisteryourPet
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver","C:\\Users\\user\\Downloads\\chromedriver_win32 (3)\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.get("http://localhost/miniproject/login.php");
        driver.manage().window().maximize();
        driver.findElement(By.id("em")).sendKeys("lakshmisunil@mca.ajce.in");
        driver.findElement(By.id("password")).sendKeys("123456");
        driver.findElement(By.id("submit")).click();
        driver.get("http://localhost/miniproject/registeryourpet.php");
        driver.findElement(By.id("petname")).sendKeys("Whisky");
        driver.findElement(By.id("petlicense")).sendKeys("A7/3457/23");
        driver.findElement(By.id("petage")).sendKeys("2");
        driver.findElement(By.id("species")).sendKeys("Dog");
        driver.findElement(By.id("breed")).sendKeys("Pug");
        driver.findElement(By.id("gender")).sendKeys("Male");
        driver.findElement(By.id("color")).sendKeys("Grey");
        driver.findElement(By.id("image")).sendKeys("C:\\xampp\\htdocs\\miniproject\\mypet2.png");
        driver.findElement(By.id("submitbutton")).click();
        String at=driver.getTitle();
        String et="My Pets";
        driver.close();
        if(at.equalsIgnoreCase(et))
        {
            System.out.println("Test Passed");
        }
        else
        {
            System.out.println("Test Failed");
        }
    }
}
```

Screenshot

```
Starting ChromeDriver 111.0.5563.64 (c710e93d5b63b7095afe8c2c17df34408078439d-refs/branch-heads/5563@{#995}) on port 29147
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Apr 01, 2023 11:01:05 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Test Passed
```

Test Report

Test Case 4

Project Name: PAWS' OWN					
Pet Registration Test Case					
Test Case ID: Test_4			**Test Designed By: LAKSHMI SUNIL**		
Test Priority(Low/Medium/High): High			**Test Designed Date: 01/04/2023**		
Module Name: Pet Registration Form			**Test Executed By : DR. BIJIMOL T. K.**		
Test Title : Register a pet with valid inputs			**Test Execution Date:01/04/2023**		
Description: Pet Registration Form					
Pre-Condition : All inputs must satisfy the given constraints					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**

1	Navigate to Login Page		Login Page	Navigated to Login Page	Pass
2	Enter valid username	username: lakshmisunil@mca.ajce.in	User should be able to Login	User logged in	Pass
3	Enter valid password	Password:123456			
4	Click on Login button				
5	Navigate to register your pet page		No error message	Pet is registered successfully	Pass
6	Enter valid pet name	Pet name: Whisky			
7	Enter valid license number:	License Number: A7/3457/23			
8	Enter valid age	Age: 2			
9	Select Species	Species: Dog			
10	Select Breed	Breed: Pug			
11	Select Gender	Gender: Male			
12	Select Complexion	Complexion: Grey			
13	Select pet photo	Choose file: C:\\xampp\\htdocs\\miniproject\\mypet2.png			
14	Click on Register Button				
15	Navigate to register your pet page				

16	Enter invalid pet name	Pet Name : 9))	Error message should be displayed	Failed to Register Pet	Pass
17	Enter invalid license number	License Number: ++			
18	Enter invalid age	Age:NULL			
19	Select NULL species	Species: NULL			
20	Select NULL breed	Breed: NULL			
21	Select NULL gender	Gender: NULL			
22	Select NULL complexion	Complexion: NULL			
23	Insert no image	Choose file: NULL			
24	Click on Register button				
Post-Condition: All inputs are validated and pet is registered successfully					

CHAPTER 6
IMPLEMENTATION

6.1 INTRODUCTION

The phase of a project where goals and plans are realised is called project implementation (or project execution). It is through project execution that ideas and plans are turned into reality. This is the obvious outcome after the evaluation, decision-making, visioning, planning, financial application, and identification of the challenge's financial resources. The project cycle's implementation stage is the most crucial because it is during this time that planned benefits are actually realised. Therefore, every level of the cycle functions as a support for this implementation stage. While assigning undertaking obligations to the assignment team in the organisation, a task must be successfully completed. The purpose of implementation are:

- To put the action plan into operation
- To manage the available resources efficiently; and
- To monitor and report on progress.

If the implementation is not well thought out or managed, confusion and mayhem may result. Implementation encompasses all of the steps used to switch from the old system to the new one. The new system could be entirely different. It could be modified to work better. In order to deliver a reliable system that meets organizational needs, proper implementation is crucial. System implementation refers to the process of actually using the built system. This comprises all the processes involved in switching from the old to the new system.

6.2 IMPLEMENTATION PROCEDURES

The act of carrying out a plan, a method, or any other design, idea, model, specification, standard, or policy is known as implementation. As a result, for something to actually occur, action must come after any previous thinking. Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. In many organizations, the software development project will be commissioned by someone who will not be operating it. Before examining the system, the user must be aware that the server programs needs to be operating on the server in order to access the results. The actual process won't happen if the server object is not active and functioning on the server.

6.2.1 User Training

User training includes instruction on how to use the device, system troubleshooting, and evaluating whether a problem is a result of the device or software. The creation of computer-based information systems must include end-user training so that employees may use the systems to solve their own

problems. The majority of user training focuses on how the technology functions. The training programs must be created so that users can quickly mobilize for the organization.

6.2.2 Training on Application Software

The user must first obtain the fundamental training in computer literacy, after which they must be taught how to operate the new application software. This will explain the fundamental principles of how to use the new system, including how the screens work, what kind of help is displayed on them, what kinds of errors are made while entering data, how each entry is validated, and how to change the date that was entered. This training may differ based on the user group and hierarchical level.

6.2.3 System Maintenance

Restoring something to its initial state is maintenance. Enhancement refers to the act of adding or changing the code to accommodate modifications made by the user specification. System enhancement increases system capability by incorporating additional requirements while system maintenance keeps the system in compliance with its original specifications. As a result, maintenance modifies the current system, enhancement adds features, and development replaces the current system. The activities that involve updating documents, testing data, and correcting mistakes made in system design and implementation make up a significant portion of system development.

6.2.4 Hosting

The project is hosted on 000webhostapp platform, top free web hosting providers offering reliable hosting services. 000WebHost includes a custom control panel as well as the possibility to connect your own domain. Web hosting allows users to store content offsite, reducing local storage costs and the associated physical footprint. It also makes it easier to build a genuinely durable web presence, with built-in advantages like back-ups for security and support.

Steps to Host a Website on 000webhostapp:

- To get free hosting, go to the 000webhost homepage and follow these steps:
- Sign up to make an account.
- Make sure you verify your email.
- Click + button to create a new website.
- Enter your details and click create.
- When you're done, click Manage Website to start building a portfolio website.

Project Link: https://pawsown.000webhostapp.com/

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

Users greatly benefit from the simplicity of PAWS' OWN, a management and reservation system for pet care items and accessories. The system under consideration aims to provide a web-based application for offering a variety of pet items and services. All of the drawbacks of the current system can be resolved by the proposed approach. The system offers all the amenities required to satisfy pet owners' wants and gives them the opportunity to schedule a visit with a veterinarian to address their animal's numerous health issues. It lessens manual labor. The current method has a number of drawbacks and numerous operational challenges. The suggested solution makes an effort to remove or significantly lessen these challenges. The user will have less workload and mental conflict thanks to the suggested system. It contains the capabilities of signing up for the system, logging in, looking for products, putting them to carts, and making purchases. Customers can schedule a consultation with the veterinarian to discuss their pets' various health issues. The suggested solution makes work easier for the user and is more time and money efficient.

.

7.2 FUTURE SCOPE

The proposed method can be made more user-friendly by creating an android application. The proposed system can be made more functional by adding the choice of providing the pets with residential care. It is possible to make a crèche service available to pet owners who are unable to care for their animals on their own. It might be made possible to post adverts about lost pets, which would be more advantageous for animal lovers.

.

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

- Raghu Ramakrishnan Johannes Gehrke, “Database Management Systems”. 2002.
- Roger S Pressman, “Software Engineering”, 1994.
- Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009.
- Pankaj Jalote, “Software engineering: a precise approach”, 2006.
- James Rumbaugh, Ivar Jacobson, Grady Booch, “Unified Modeling Language Reference Manual”, 2004.

WEBSITES:

- https://www.w3schools.com/
- https://www.flipkart.com/
- https://headsupfortails.com/
- https://www.petsy.online/
- https://www.geeksforgeeks.org/
- https://www.tutorialspoint.com/uml/uml_overview.htm

CHAPTER 9
APPENDIX

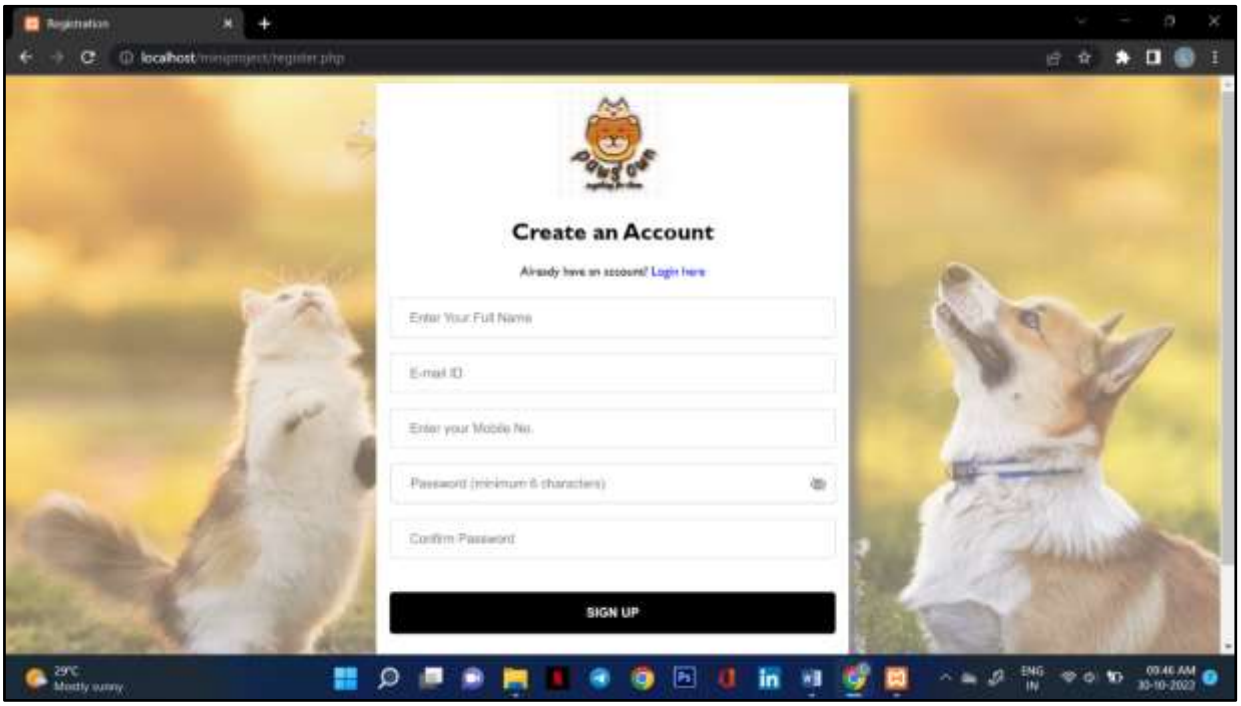
9.1 Sample Code

9.1.1 Registration

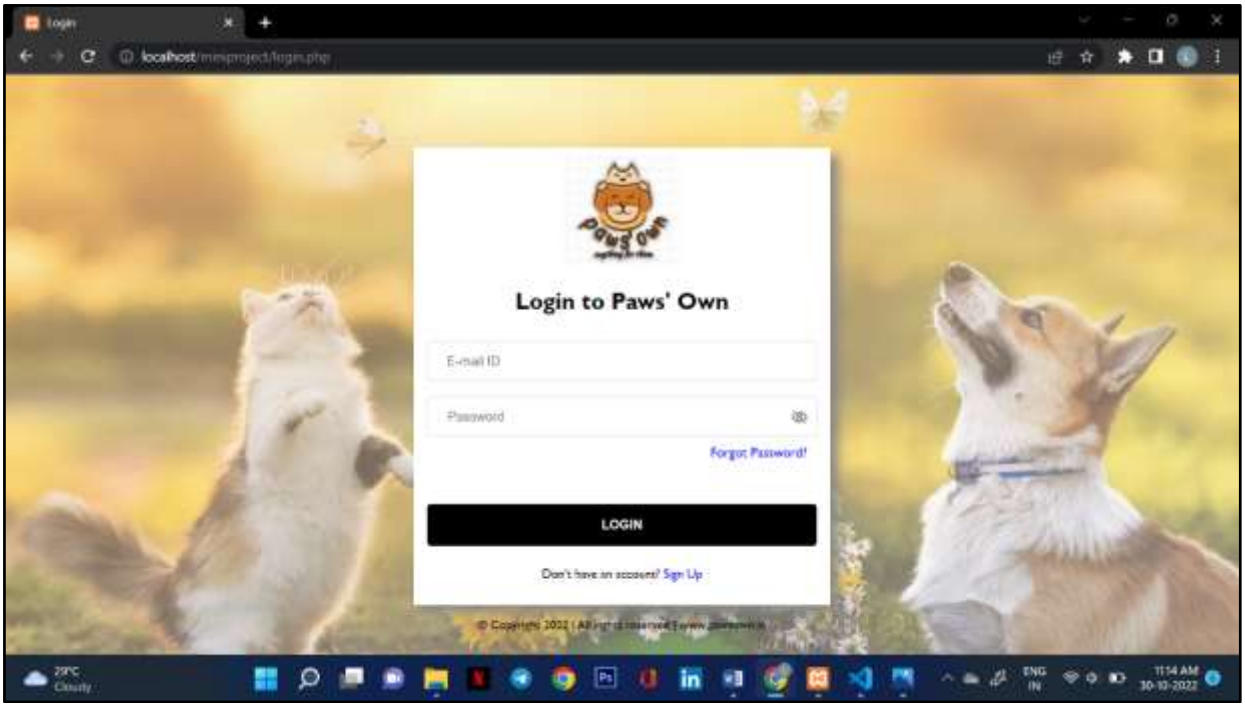
```
<?php
include "connection.php";
session_start();
if(isset($_POST["submitdata"]))
{
$custname=$_POST["cusname"];
$_SESSION["cus"]=$custname;
$custemail=$_POST["cusemail"];
$custphone=$_POST["cusphone"];
$custpassword=$_POST["cuspassword"];
$encryptedpassword= md5($custpassword);
$data="INSERT INTO tbl_userdetails VALUES
(NULL,'$custname','$custemail','$custphone','Customer','$encryptedpassword',
1,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL)";
try
{
    if($conn->query($data)===true);?>
    <div id="snackbar">Account Created! Please Login to Continue</div>
    <script> var x = document.getElementById("snackbar");
    x.className = "show";
    setTimeout(function(){ x.className = x.className.replace("show", ""); },
3000);</script>
<?php
}
catch(Exception)
{
    <div id="snackbar2">Email ID/Phone already registered</div>
    <script> var x = document.getElementById("snackbar2");
  x.className = "show";
  setTimeout(function(){ x.className = x.className.replace("show", ""); },
3000);</script>
<?php
}}
?>
 <?php
 if(isset($_GET["status"]))
 {
  if(($_GET["status"])==1)
  {
     ?>
     <div id="snackbar3">Access Denied! Invalid Email ID/ Password</div>
            <script> var x = document.getElementById("snackbar3");
          x.className = "show";
          setTimeout(function(){ x.className = x.className.replace("show", ""); },
3000);</script>     <?php}}?>
```

1.2 Screenshots

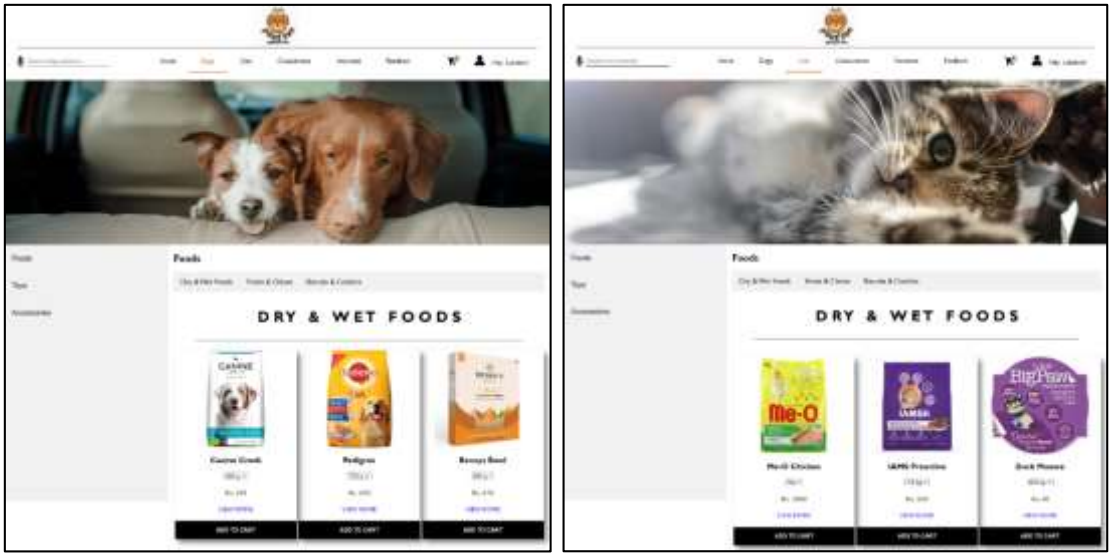
1.2.1 Registration



1.2.2 Login



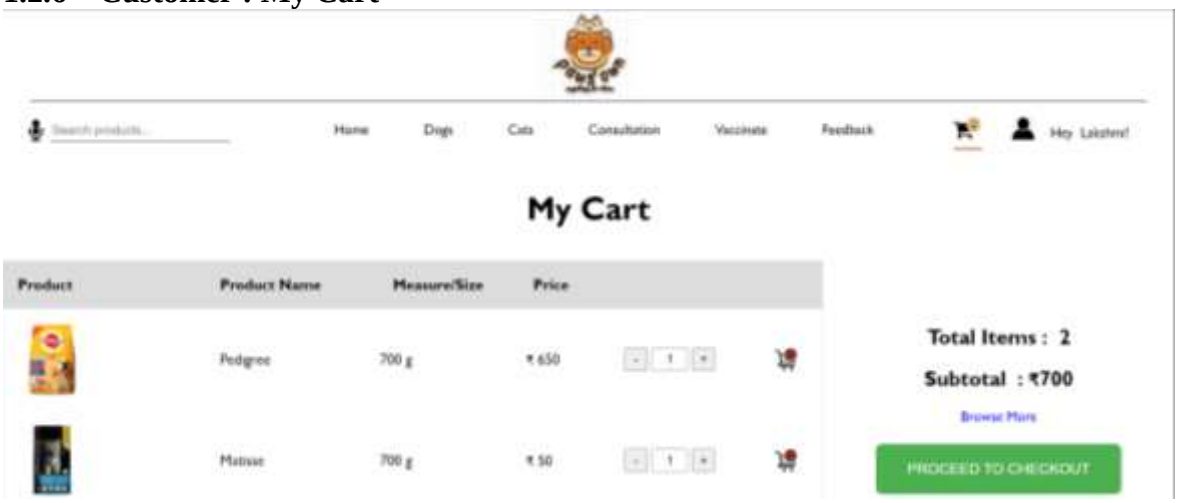
1.2.3 Customer : Home Page – Dogs & Cats



1.2.4 Customer : Appointment Form

1.2.5 Customer : Vaccination Booking Form

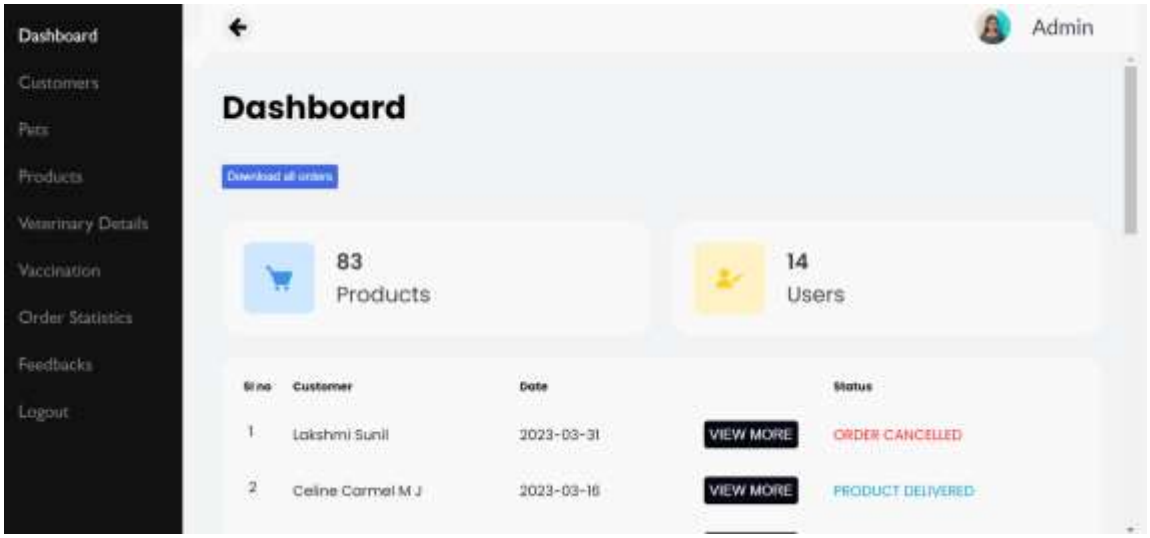
1.2.6 Customer : My Cart



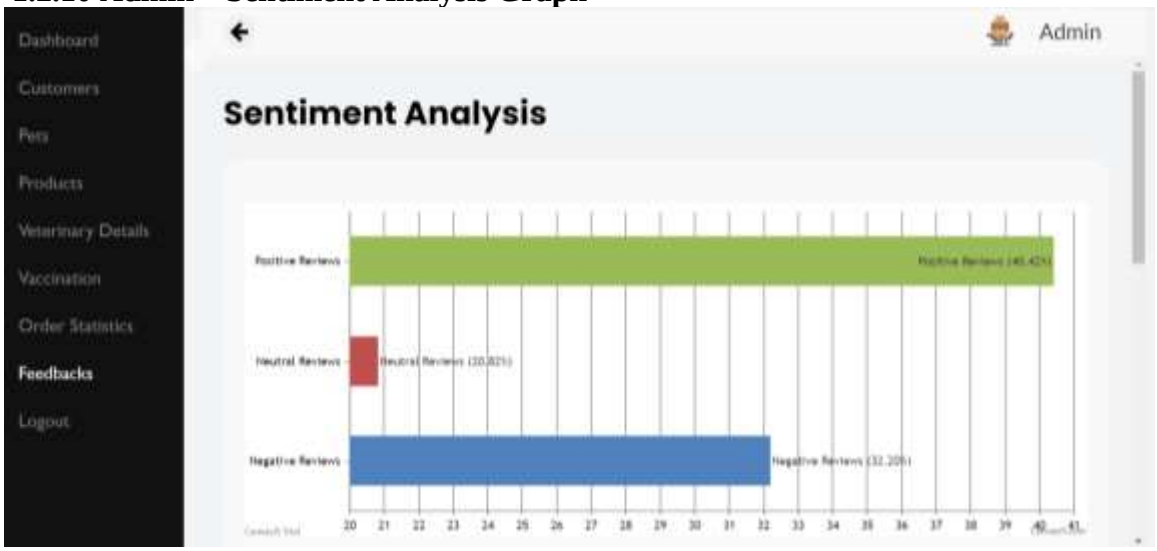
1.2.7 Customers : Payment

1.2.8 Customer : Pet Registration Form

1.2.9 Admin – Dashboard



1.2.10 Admin – Sentiment Analysis Graph



1.2.11 Vaccine Center - Dashboard