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]

2018-2023

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 and indebtedness 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 store that specializes in selling products and accessories for pets is known as a pet care store. These types of retail businesses offer a range of resources for the care of pets, and they also sell a variety of animal supplies and accessories. 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
RDBMS	-	Relational Database Management System
PHP	-	Hypertext preprocessor

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The management system for pet care products and accessories is an online application that helps to manage the operations of a physical pet store. Its proposed approach fulfills all the necessary requirements for pet owners to provide optimal care for their animals. This particular web-based application enables customers to purchase pet supplies and view all relevant information about the store. Additionally, it includes an online application form for scheduling consultations with veterinarians to discuss the health of their dogs, as well as the ability for pet owners to schedule vaccination appointments from the comfort of their own homes.

1.2 PROJECT SPECIFICATION

The pet care products and accessories management system is an online application that enables pet owners to purchase products and request services. The system includes an online application form for customers to schedule appointments with veterinarians and vaccinations for their pets. Furthermore, customers can search or browse through a range of products provided by the system. The system also offers a vaccination plan and a schedule for pet healthcare to help customers maintain their pets' well-being. The web-based system is essential for storing customer data and conducting everyday business for the pet care products and accessories store in a more organized manner. This will provide clients with easy and accessible information about the system. The system consists of four modules, which 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 research or analysis refers to the process of examining data, identifying issues, and proposing solutions for system improvements to achieve a procedure or business's objectives. In order to achieve success, the systems and processes must be designed with collaboration between users and system developers. To identify the areas of concern, the system is thoroughly examined, inputs are acknowledged, and the system is viewed as a whole. Solutions are provided as recommendations, and the concept is evaluated and adjusted based on user feedback. During the process of solving problems related to system development, it is crucial to establish clear and efficient communication between the developers and users of the system. The first step in any system development process should involve system analysis or research. The current system is thoroughly examined and evaluated, with a system analyst acting as an investigator to scrutinize the system's internal workings. The inputs to the system are identified, and the system is considered as a complete entity. The various processes within the system can then be linked to the outputs of the organization.

To comprehend the problem and identify the relevant variables, system analysis involves the synthesis and analysis of various components, followed by selecting the most feasible course of action. Various techniques such as interviews and surveys are employed to thoroughly investigate the process, and data from multiple sources is evaluated to reach a conclusion about how the system works. This is referred to as the current system. After a thorough analysis of the current system, the designer assumes the responsibility of identifying and proposing solutions to address the identified problem areas. An analytical comparison of the proposed solutions and the existing system is conducted to select the optimal solution. The proposed solution is presented to the user for their feedback, and revisions are made based on their input until they are satisfied with the proposed solution. The process of gathering and analyzing data for future system studies is referred to as preliminary research. It refers to the process of acquiring and analyzing data to be used for future system studies. It involves close coordination between system users and developers and includes feasibility studies to estimate system activities and determine efficient research and analysis tactics. Collaboration between system users and developers is critical for problem-solving in initial research. Feasibility studies are conducted as part of preliminary research to roughly estimate the system activities, which aid in selecting the best methods for effective 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

PAWS' OWN's major objective is to develop an internet application that would allow customers to purchase numerous pet case products and accessories without having to go to a physical 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

Feasibility studies are conducted to provide an objective evaluation of a project's potential success. It involves a systematic analysis of the project's technical, economic, and operational aspects to determine whether the project is feasible and worth pursuing. It cover a broad range of factors that impact the project's success, including technical, economic feasibility, and operational feasibility. Feasibility studies identify potential risks associated with the project, such as technical difficulties, market changes, or unforeseen obstacles. Feasibility studies include a cost-benefit analysis, which helps determine whether the benefits of the project outweigh its costs. This analysis takes into account both tangible and intangible benefits and cost. It has the following features:

4.1.1 Economic Feasibility

We need to analyze the costs and benefits to support the development of the system. We want to make sure we prioritize the project that will give us the best results as quickly as possible. One of the factors to consider is the cost of setting up a new system. During the initial probe, some important financial questions were raised such as:

- How much would it cost to do a thorough system investigation?
- What would be the expense of the hardware and software?
- What kind of potential benefits could be gained in terms of reduced costs or fewer costly mistakes?

Since the proposed system was developed as a part of a project, there are no additional manual costs associated with it. The costs associated with the system can be categorized into three groups: system costs, development costs, and hosting costs. The employment of open-source software during the development process resulted in a cost-effective system, as indicated by all calculations. By utilizing cost-effective resources, the system was successfully established at a low cost.

4.1.2 Technical Feasibility

Before moving forward, we need to technically assess the system. A design overview of the system's requirements such as input, output, programs, and procedures should be used for assessing its feasibility. In addition, it is necessary to examine the required equipment, the construction procedures, and the operational methods for the proposed system once the design and outline have been established. While conducting the investigation, technical issues were raised, such as:

- Is the current technology adequate for supporting the proposed system?
- Can the system be scaled up if it is developed?

The creation of the project should guarantee that it operates effectively and performs as expected within the established boundaries. The ability of a newer version of the software to operate with an older one indicates that the system can continue to be used even if the technology becomes outdated at some point. Therefore, there are few limitations for this task. The system's front and back ends have been developed using HTML and CSS, making it feasible to create. Additionally, the system exhibited strong performance using an Intel i7 core processor, 8GB RAM, and 1TB hard drive.

4.1.3 Behavioral Feasibility

PAWS' OWN includes the following questions:

- Do we have enough support for the users?
- Is there a risk of potential harm with this new system?

If all behavioral factors are taken into account, this project appears to be behaviorally feasible. When it's completed, it should be able to reach its goals. 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 proposed system aims to enhance the business's profitability and sales by reaching the maximum number of customers at the appropriate time.

2. Which individuals utilize the system?

   The system is utilized by three groups of individuals: the admin, veterinarian, and 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 manufacturer advises that this vaccine should be given to children starting at 8 weeks old. Administer the first dose at 6–8 weeks of age, with a second dose to follow 2–4 weeks later.

Then, repeat this process annually or more frequently in high-risk animals that are not covered by an annual booster

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

   The listed items consist of salmon, bread, eggs, spinach, chicken, turkey, green beans, and 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, which stands for PHP: Hypertext Preprocessor, is a server-side scripting language that is specifically designed for website development. It is an open-source programming language that can be downloaded and used for free, and it is easy to learn and use. The file extension for PHP files is ".php". Rasmus Lerdorf is the individual who inspired both the original PHP and its subsequent versions. PHP is an interpreted language, so there is no need for a compiler. PHP commands can be embedded directly into the HTML source document, eliminating the need for an external file to handle data. PHP has evolved to include a command line interface and is incompatible with the GNU General Public License (GPL) due to usage limitations. PHP can be deployed for free on most web servers, and it can also be used as a standalone shell on a wide range of platforms and operating systems. It can be combined with various databases, including Oracle, Microsoft SQL Server, MySQL, PostgreSQL, Sybase, and Informix. PHP is robust enough to support user access restrictions and content management systems such as WordPress. PHP codes are executed on the server, unlike HTML codes, which are directly rendered in the browser, making PHP distinct from client-side languages such as HTML.

3.3.2 MySQL

MySQL is a software used for managing relational databases (RDBMS) that allows for the storage, organization, and retrieval of data. This particular software is open-source and has various applications. It is created, distributed, and supported by Oracle Corporation. MySQL is designed to function using a client-server system, and can work with several programming languages, such as Java, PHP, and C++. Its capacity for scalability, speed, and dependability has made it popular in various fields like finance, healthcare, and e-commerce. MySQL offers features for data encryption, replication, and high availability to secure and make data accessible. Moreover, it presents diverse storage engines, such as InnoDB, MyISAM, and NDB, which enable users to choose the best-suited one for their specific necessities.

- A database management system is MySQL: MySQL is a database management system that is utilized for managing systematic collections of data, also known as databases. Databases can take various forms, from simple grocery lists to large amounts of data in business networks. To add, access, and process data within a computer database, a database management system like MySQL Server is required. These systems are crucial to computing and can be used as standalone programs or as a part of other programs, as computers are proficient in processing large volumes of data.

- MySQL, a type of relational database, uses tables to organize data instead of storing all of it in one location. The database structures are stored in optimized physical files to enhance speed. A relational database system's logical model consists of several objects such as tables, views, databases, rows, and columns, which provide a versatile programming environment.

- MySQL is a type of open-source software that allows users to utilize and adjust the source code without incurring any expenses. The software is governed by the GNU General Public License (GPL), which specifies the restrictions and authorizations under particular conditions. This allows for the review and modification of the MySQL source code to suit individual requirements. MySQL's commercially licensed version can be purchased by those who are uncomfortable with the GPL's restrictions or who wish to include MySQL code in a for-profit application.

- The MySQL software is made up of several components, including a server, client programs and libraries, administrative tools, and APIs. The server supports various back-ends and is capable of multi-threading. It can also be utilized as a library, allowing for the creation of smaller and faster standalone applications by linking it to a program.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

To create any product or system, the first step is the design process, which is a creative process. Designing is essential for the efficiency of the system. Design is the process of using various techniques and ideas to specify the details of a process, device, or system to make it possible for it to be physically constructed. In software engineering, design is a crucial technical aspect of the development process, regardless of the approach used. The responsibility of system design is to create the necessary architectural specifications to build a product or system. During the design phase, this program has undergone rigorous refinement of its efficiency, performance, and accuracy to ensure optimal results, as is common in any systematic approach. In this phase, a document intended for users is transformed into one that programmers or database staff can use. The system design development process consists of two stages: logical design and physical design.

4.2 UML DIAGRAM

UML is a widely used language for software system design, visualization, building, and documentation. They are graphical representations used for modeling software systems, processes, and structures. The Object Management Group (OMG) developed UML, and it was first defined as UML 1.0 in January 1997. UML is a graphical language that is separate from programming languages such as C++, Java, and COBOL, and is utilized to create software blueprints. UML is a general-purpose visual modeling language that is used for system visualization, specification, construction, and documentation, not only for software systems but also for non-software-based systems like process flows in a manufacturing facility. Although UML is not a programming language, it can be utilized to generate code in various languages using UML diagrams. UML is directly connected to the analysis and design of object-oriented systems. It has been standardized and is currently an official standard of the OMG. A comprehensive UML diagram includes all the system elements and relationships, and its visual appearance is critical. UML is composed of nine diagrams, and adding all the supplementary components completes the diagram. Each type of diagram in UML represents a specific aspect of the system being modeled. For example, a class diagram shows the classes and their relationships in a system, while a sequence diagram depicts the interactions between objects in a specific scenario. UML diagrams are used to document, communicate, and analyze software and system designs, as well as to help identify potential issues and improve system performance. The nine different UML diagrams are:

- 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 type of UML diagram used to visualize the interaction between various components of a system. It is used to identify, define, and organize system requirements by outlining different scenarios that the system must handle. Use cases describe a sequence of actions that a system performs to achieve a specific goal for an actor (a user, a device, or another system) interacting with it. The use case diagram provides a high-level overview of the system's functionality and illustrates the actors, use cases, and the relationships between them. The actors are represented by stick figures, and the use cases are represented by ovals. The relationships between them are shown using arrows. Use case diagrams can help software developers, business analysts, and other stakeholders to communicate and understand the system's functionality in a clear and concise way. They are often used in the early stages of software development to ensure that all system requirements are captured and to help identify any potential issues with the design. A use case diagram is made up of four elements. They are:

- The system boundary that outlines the scope of the system, the actors who represent individuals and their roles in the system.
- The use cases that describe specific interactions between actors and the system.
- The actors who represent individuals and their roles in the system.
- The relationships that illustrate the connections between actors and use cases.

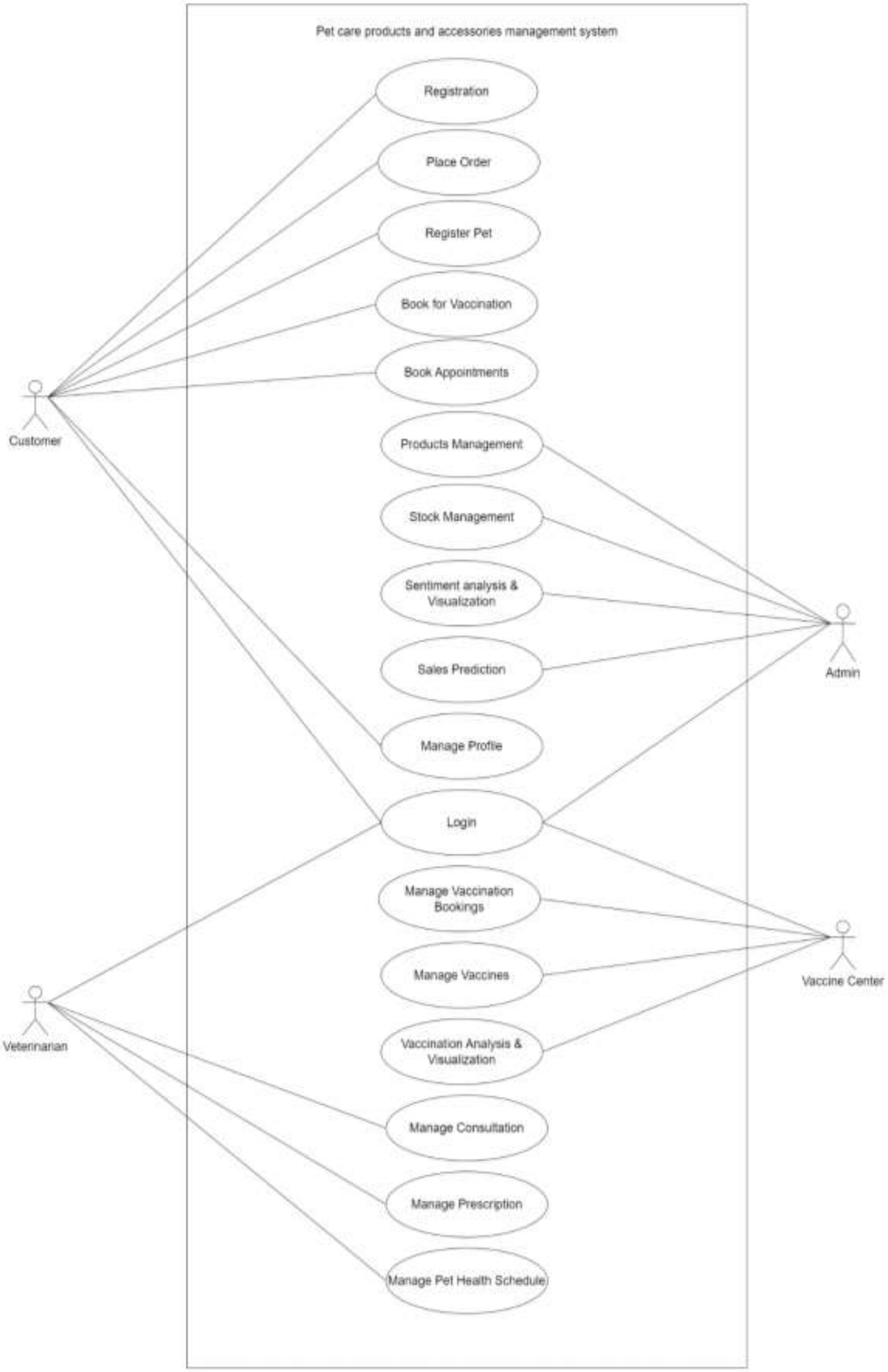


Figure 4.2.1.1: Use Case Diagram

4.2.2 SEQUENCE DIAGRAM

A sequence diagram is a visual representation that shows the interactions and order of events between multiple entities. It is a specific type of interaction diagram and displays the sequence of messages passed between objects in chronological order. The diagram is used to illustrate the objects and classes involved in a given scenario and the messages exchanged between them. Typically, sequence diagrams are associated with use case realizations in the Logical View of the system being developed. They are sometimes called event diagrams or event scenarios. In sequence diagrams, concurrent processes or objects are displayed as vertical parallel lines, called lifelines, and the messages exchanged between them are represented by horizontal arrows in the order they occur.

.

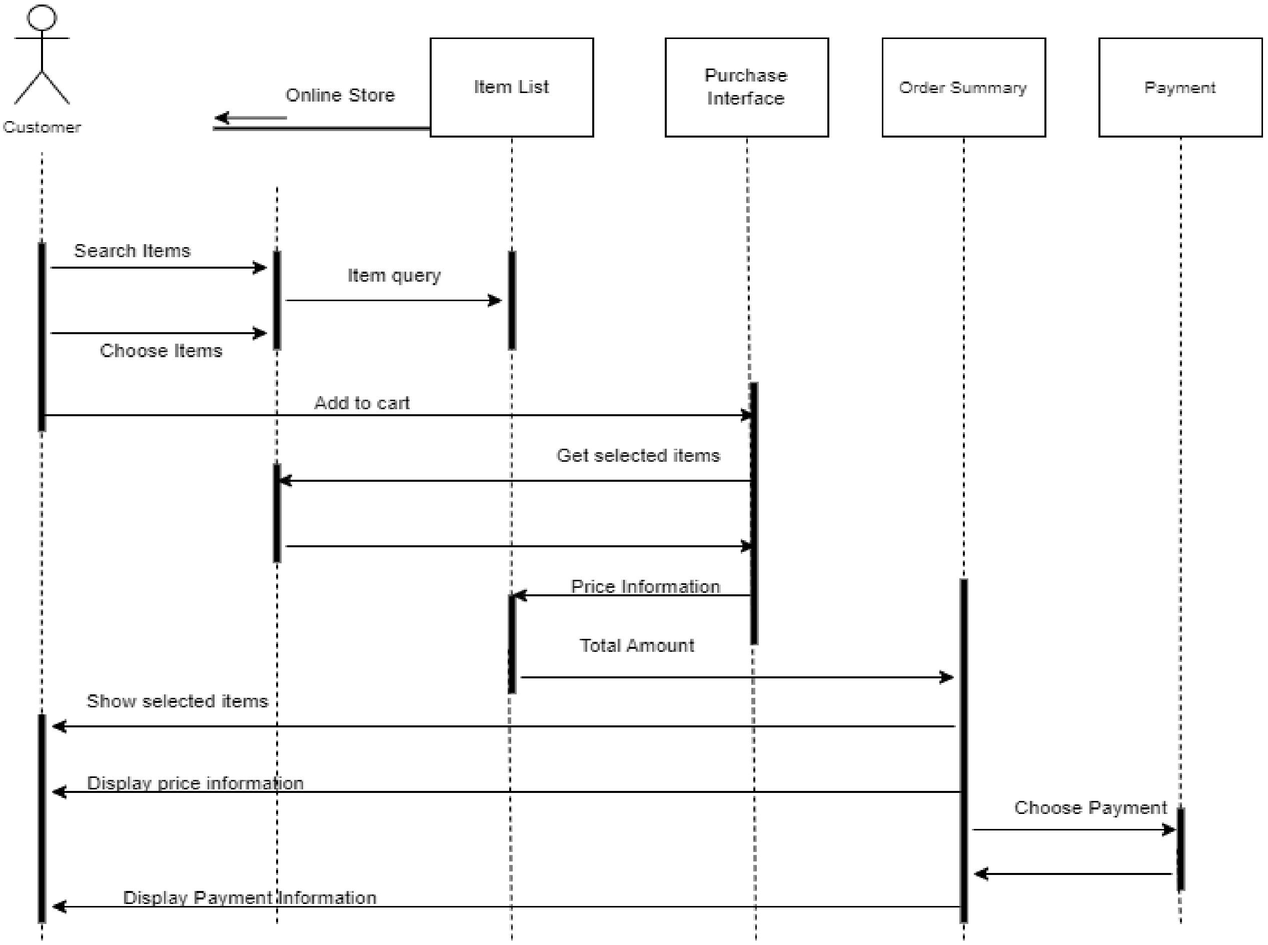


Figure 4.2.2.1: Sequence diagram for the proposed system

4.2.3 State Chart Diagram

A state chart diagram is a graphical representation that illustrates the various statuses of system components. Each state corresponds to a specific object or component of the system. State machines, which identify an object's different states and manage them in response to internal or external events, are shown in state chart diagrams. Multiple states exist throughout an object's lifetime, and these states are altered by events. State chart diagrams are employed to represent reactive systems, which are systems that react to internal or external events. A state chart diagram portrays the transfer of control between states, with each state representing a condition in which an object exists and is altered by an event. The primary goal of a state chart diagram is to model the lifecycle of an object from creation to termination. State chart diagrams are also useful for both forward and reverse engineering of a system, despite being mainly used to model reactive systems.

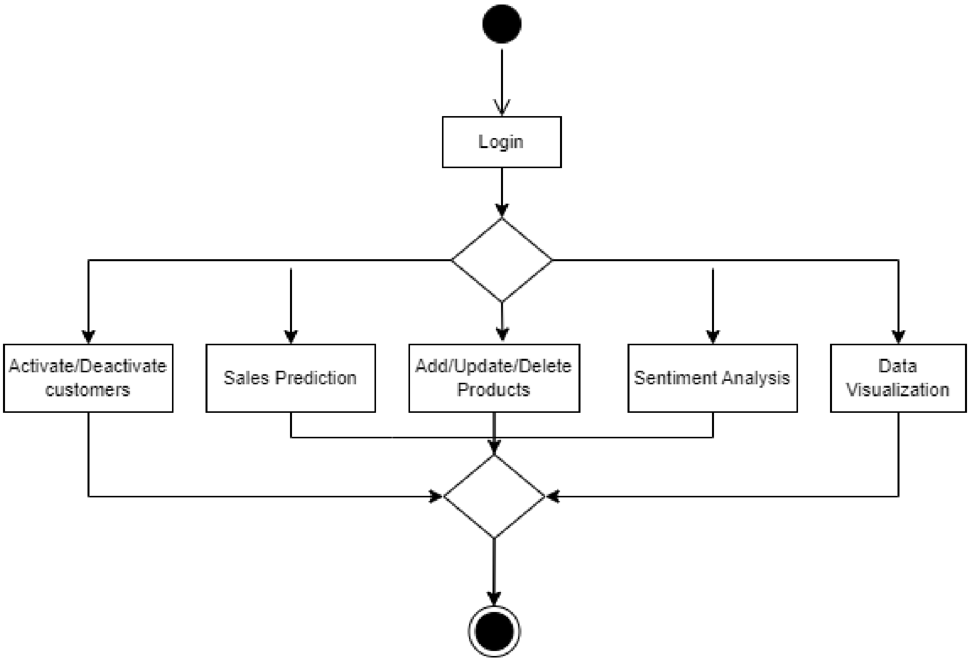


Figure 4.2.3.1: State Chart Diagram for Admin

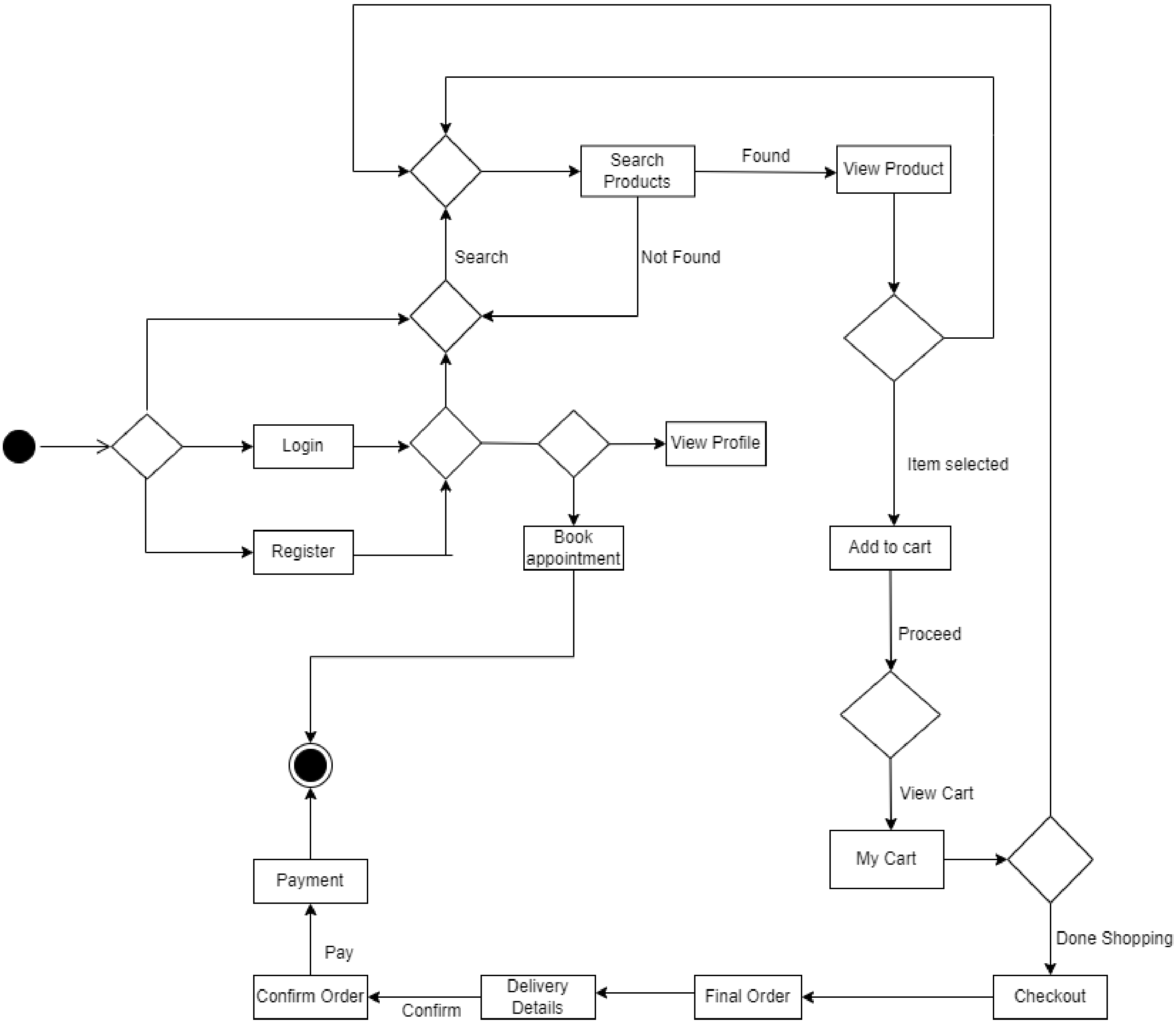


Figure 4.2.3.2: State Chart Diagram for Customer

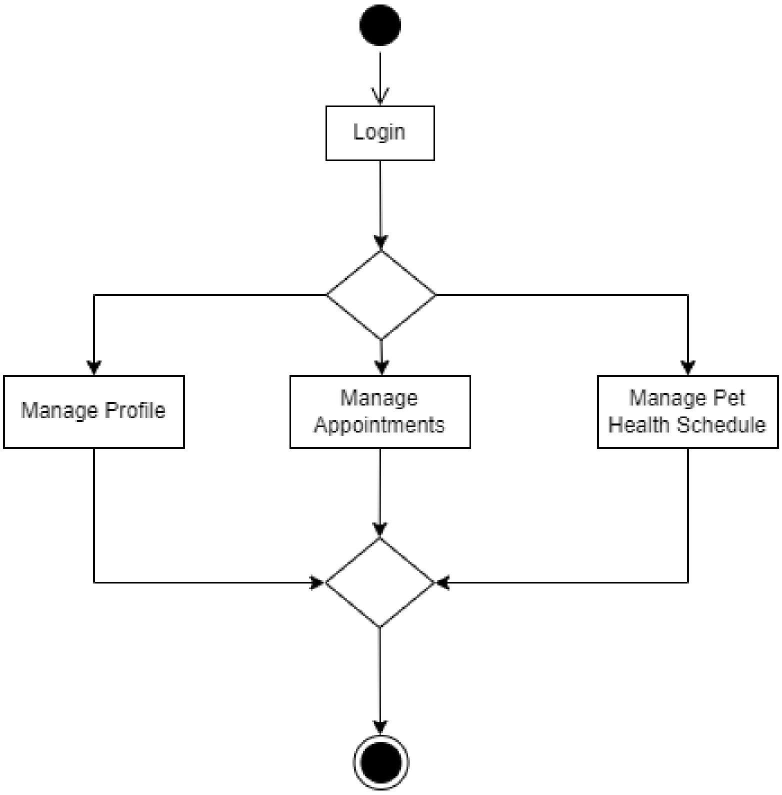


Figure 4.2.3.3: State Chart Diagram for Veterinarian

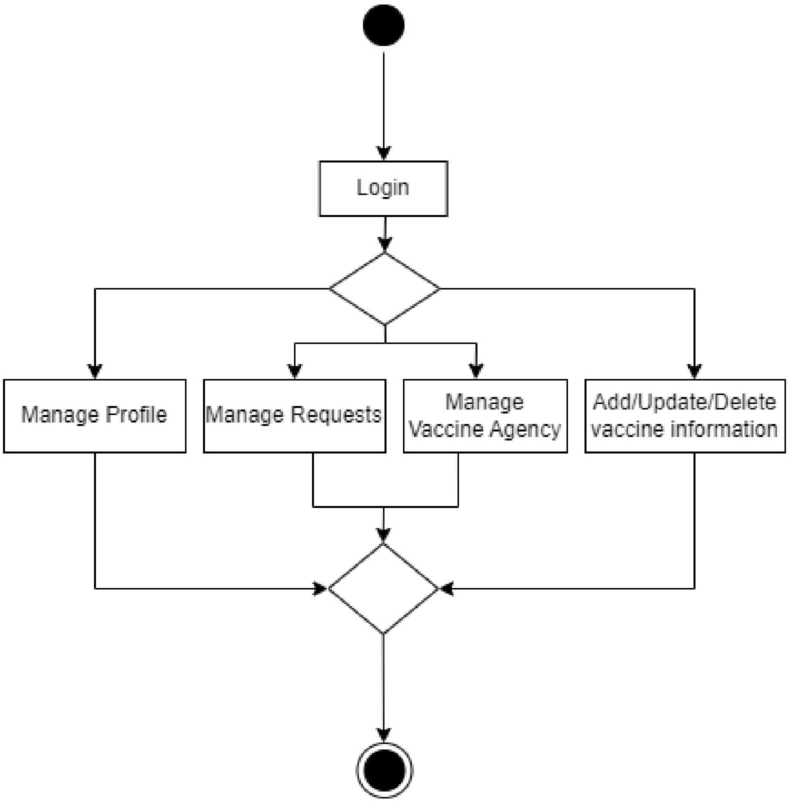


Figure 4.2.3.4: State Chart Diagram for Vaccine Center

4.2.4 Activity Diagram

The activity diagram is a visual representation of a system's dynamic aspects that shows how different activities are connected to one another, similar to a flowchart. The activities refer to the operations that the system performs, and the connections between them are represented by control flow, which can be concurrent, parallel, or branched. The diagram employs different constructs, such as fork and join, to manage various types of flow control. It is utilized for both forward and reverse engineering of executable systems and to visually demonstrate the dynamic nature of a system. Unlike other diagrams, the activity diagram does not illustrate message flow between activities. While an activity diagram can replace a flowchart in some situations, it is not the same as a flowchart, despite their similarities.

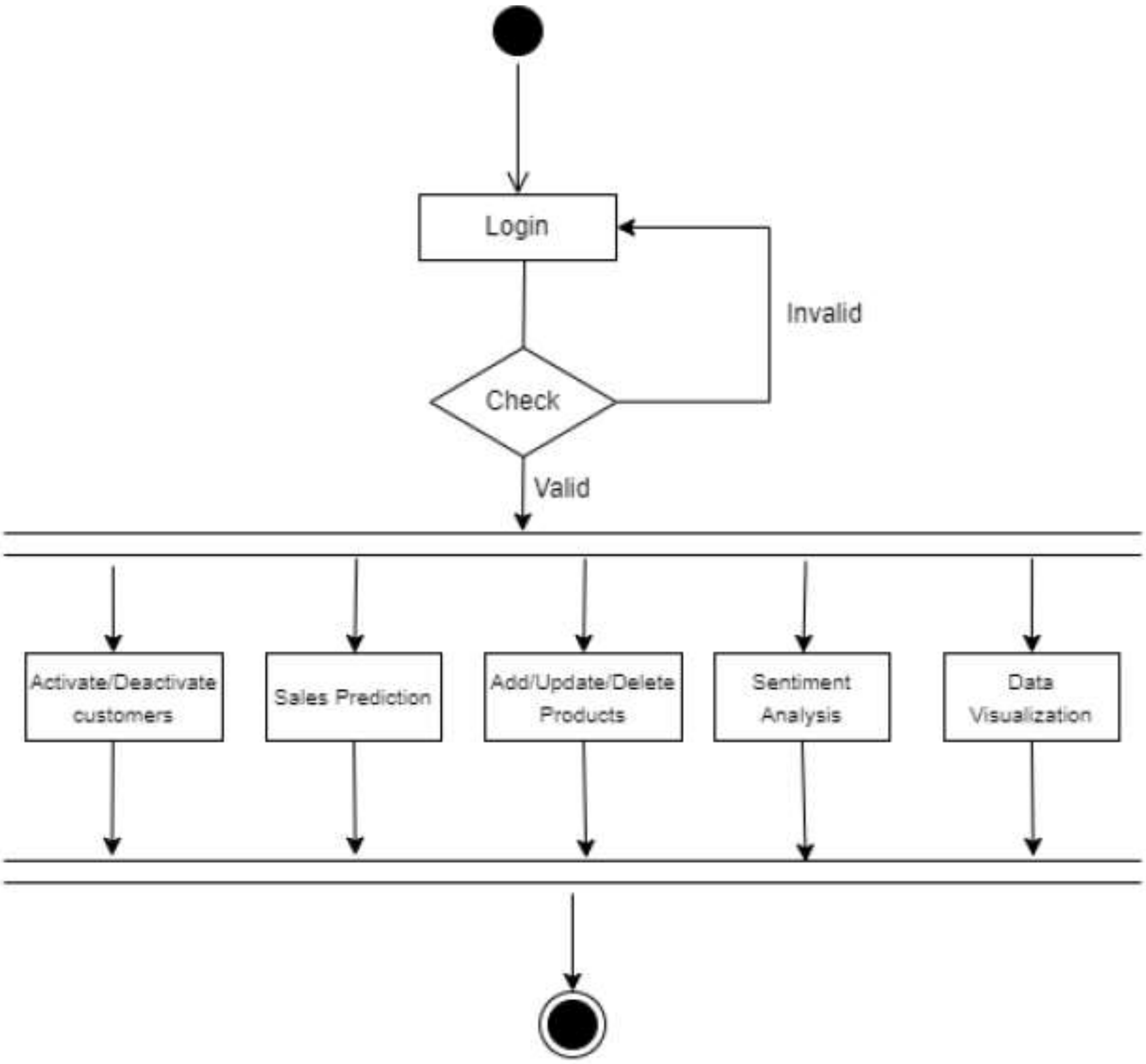


Figure 4.2.4.1 : Activity Diagram for Admin

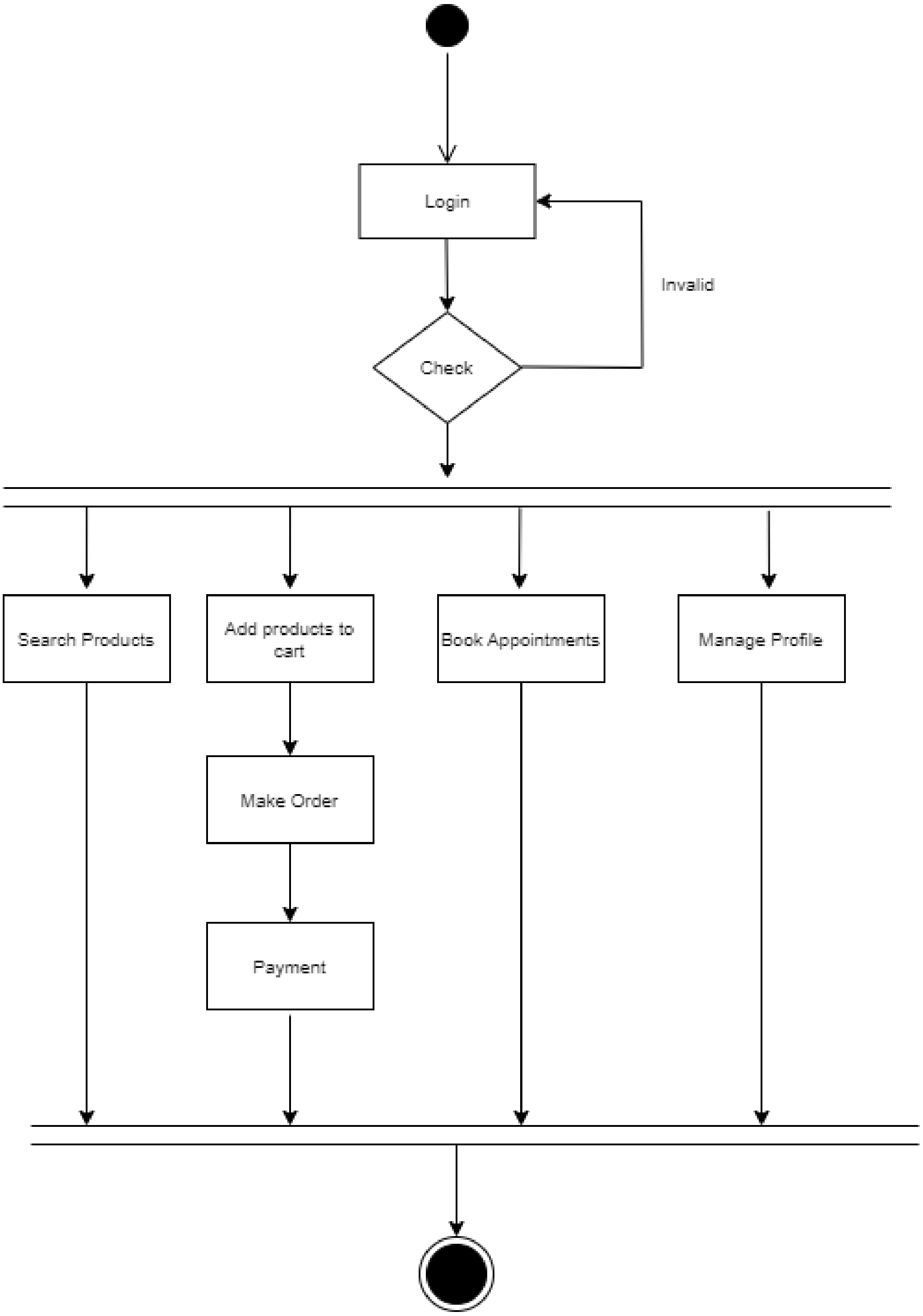


Figure 4.2.4.2: Activity Diagram for Customers

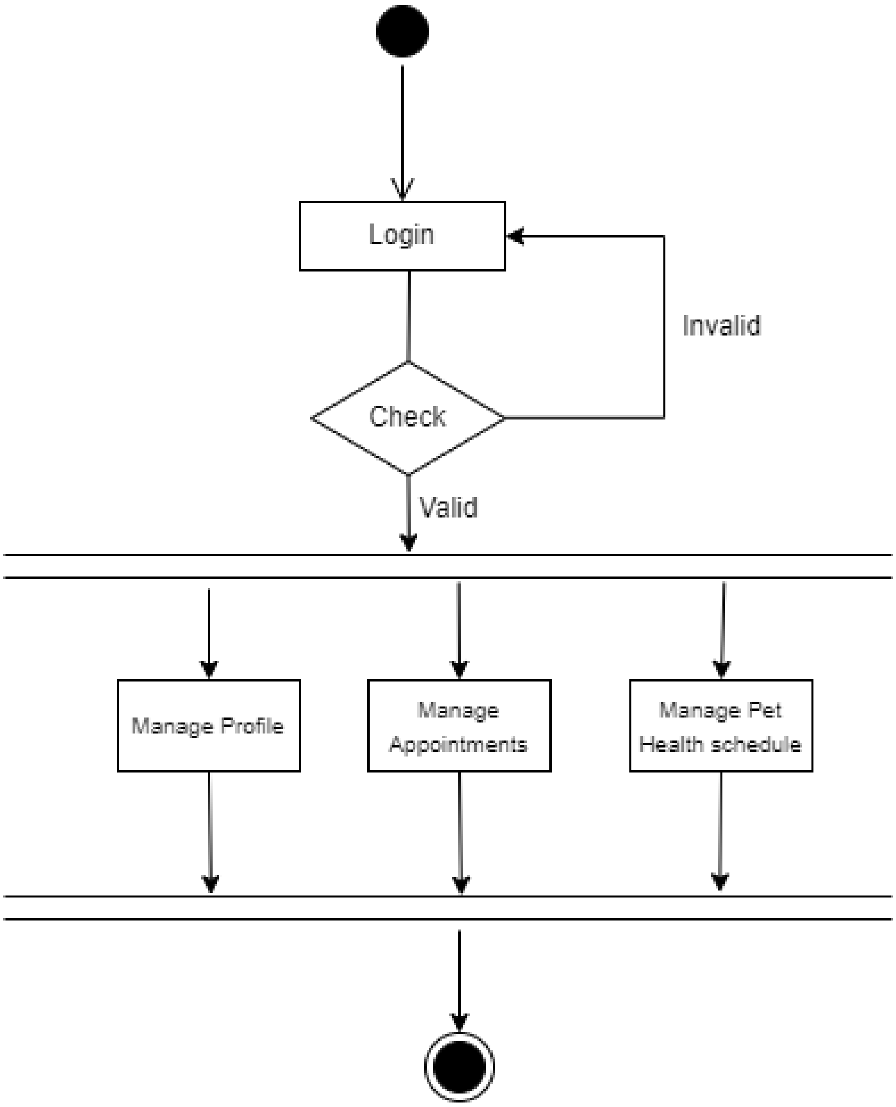


Figure 4.2.4.3: Activity Diagram for Veterinarian

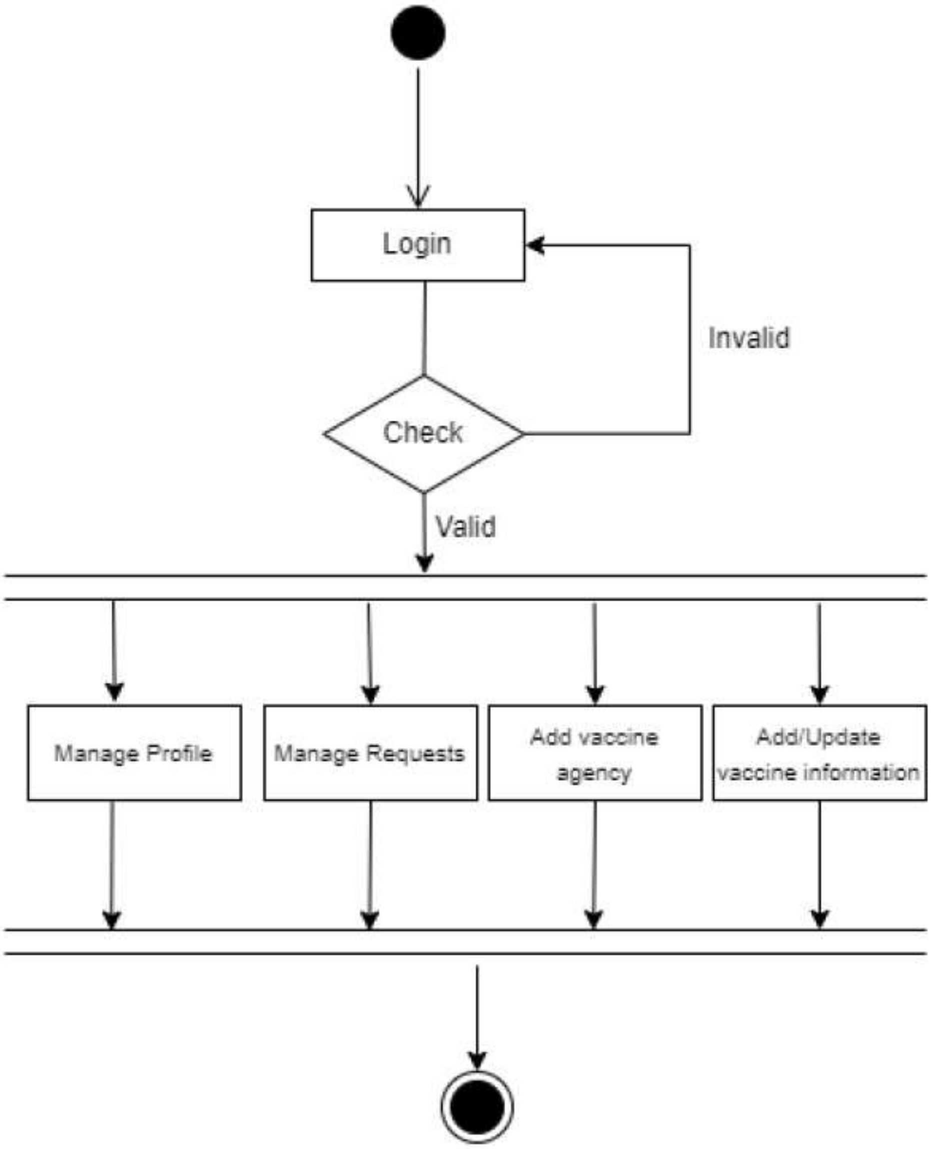


Figure 4.2.4.4: Activity Diagram for Vaccine Center

4.2.5 Class Diagram

A class diagram is a static diagram that shows a static view of an application. These diagrams are used to visually represent and document various elements of a system and are useful for creating executable code for software applications. The diagram presents the attributes and operations of a class, as well as the constraints that are imposed on the system. Class diagrams are particularly important in object-oriented systems modelling since they can be translated directly into object-oriented programming languages. The diagram displays a collection of classes, interfaces, relationships, collaborations, and constraints and is also known as a structural diagram.

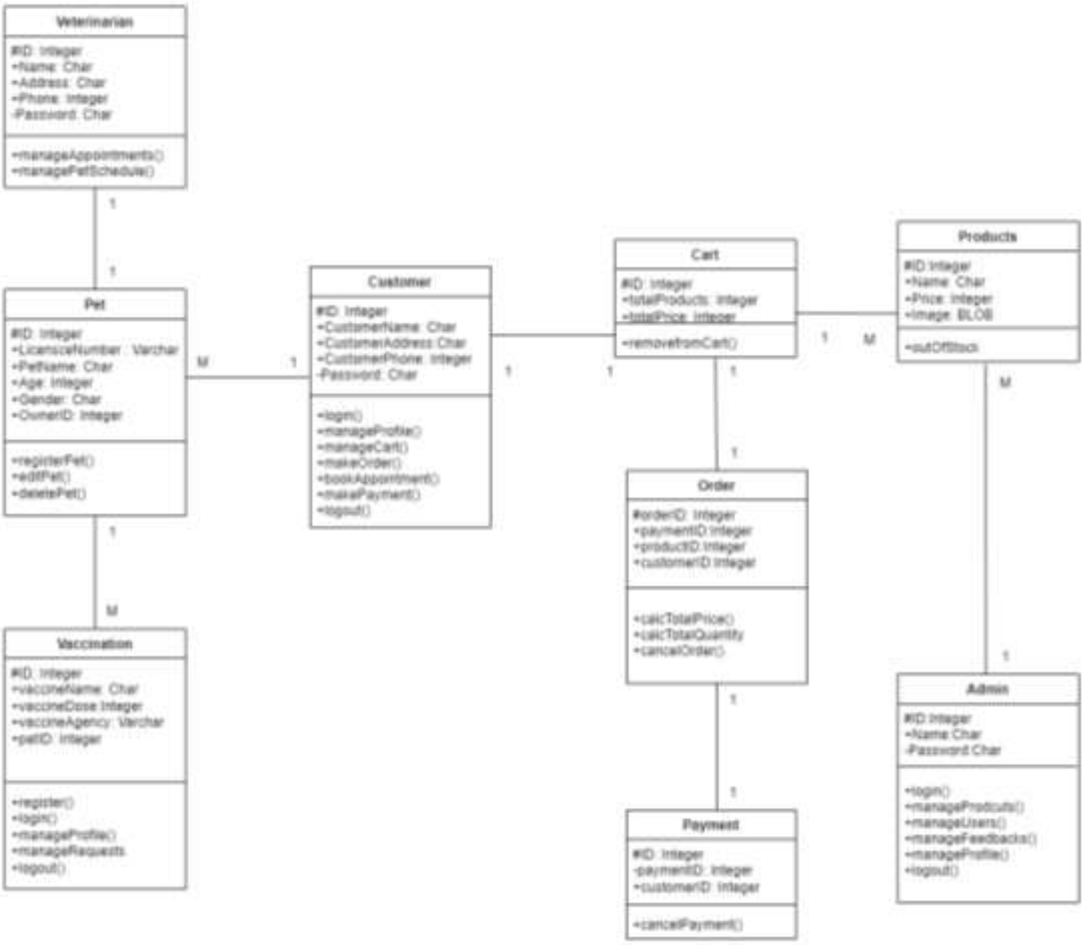


Figure 4.2.5.1 Class diagram for the proposed system

4.2.6 Object Diagram

Object diagrams are a type of UML diagram that depict a specific instance or snapshot of a system. They are derived from class diagrams, which provide a static view of the system. Object diagrams and class diagrams share similar concepts and principles, but while class diagrams show the system's structure, object diagrams represent a specific moment in time. Object diagrams are used to display the relationships between a group of objects and their connections, while class diagrams are used to describe the attributes, operations, and relationships between classes. Class diagrams must be created before object diagrams since the latter are based on the former. The key goals of object diagrams are:

- To illustrate the relationships among objects in a system
- Provide a static view of an interaction
- Help understand the behavior and relationships of objects from a practical perspective

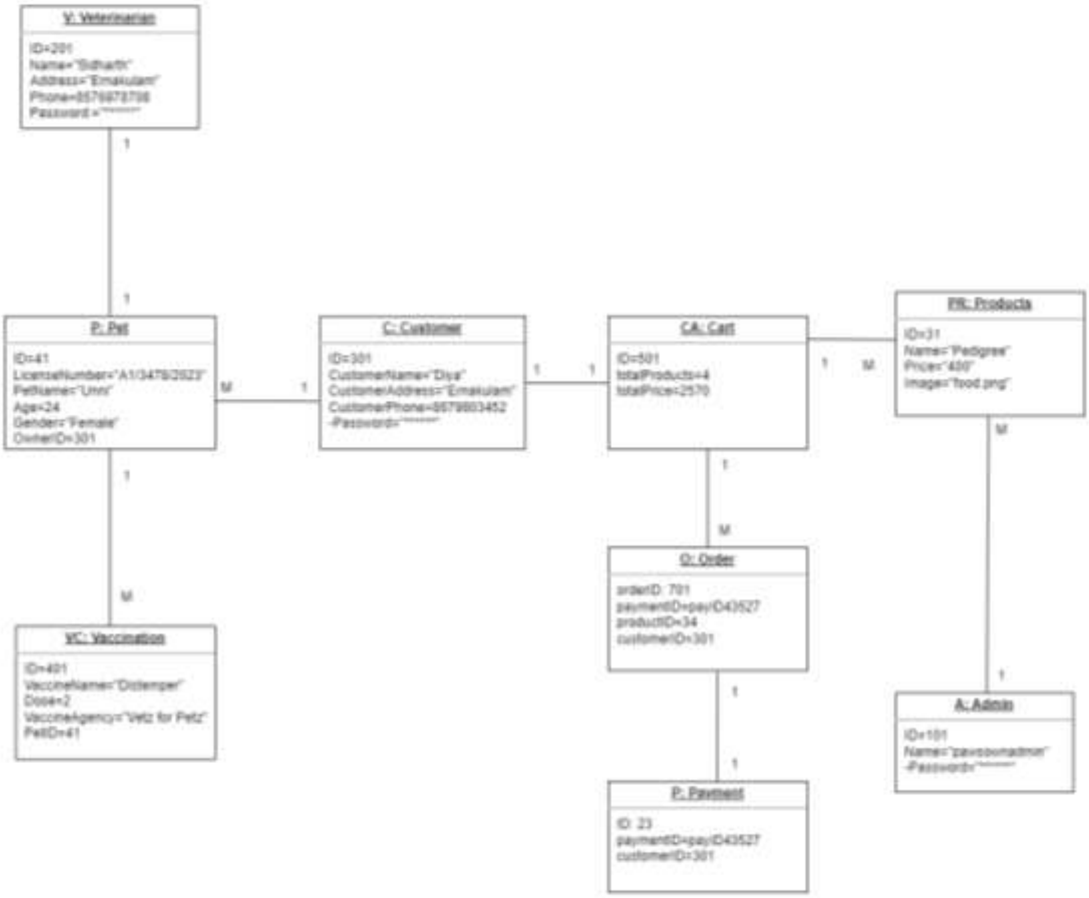


Figure 4.2.6.1: Object diagram for the proposed system

4.2.7 Component Diagram

A UML diagram known as the component diagram has a different function compared to other diagrams. It does not illustrate how the system works, but instead depicts the physical components that contribute to the system's functionality. Component diagrams are useful in representing the concrete parts of a system, such as files, libraries, and packages. These diagrams provide a static view of the system's implementation by demonstrating the arrangement of components at a particular moment. Since a single diagram cannot capture the entire system, a set of diagrams is utilized. To summarize, the component diagram serves the following purposes:

- To provide a visualization of the system components.
- To facilitate the creation of executable code through forward and reverse engineering techniques.
- To describe the structure and connections between the components.

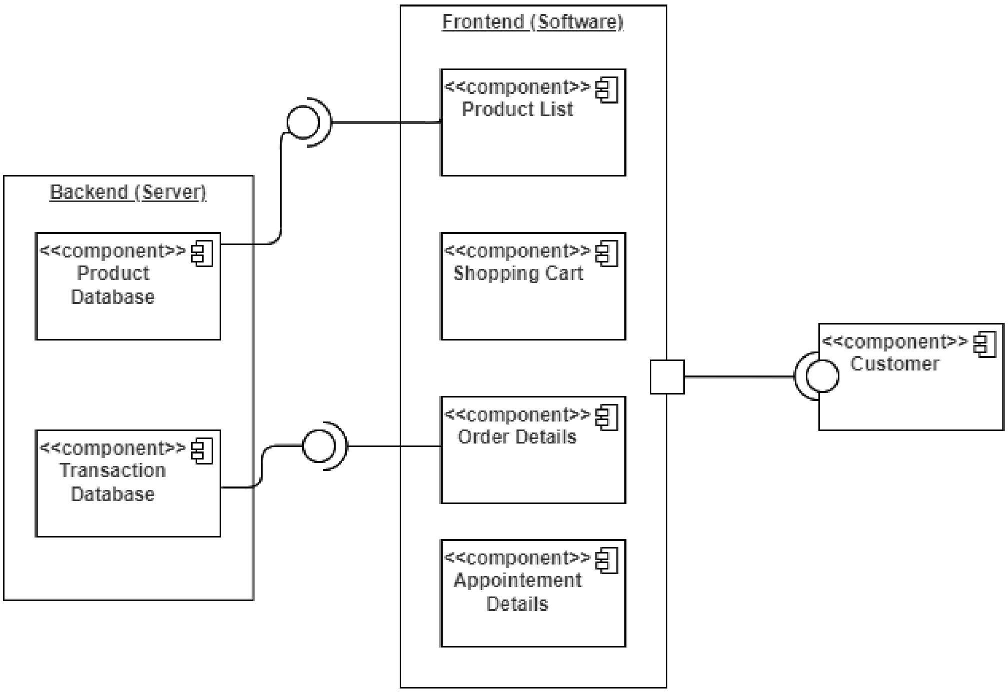

Frontend (Software)
<<component>> Product List
<<component>> Shopping Cart
<<component>> Order Details
<<component>> Appointement Details
Backend (Server)
<<component>> Product Database
<<component>> Transaction Database
<<component>> Customer

Figure 4.2.7.1: Component diagram for the proposed system

4.2.8 Deployment Diagram

A UML diagram type known as deployment diagrams is used to depict the physical configuration of a system and the installation locations of its software components. The primary goal of a deployment diagram is to visualize the static deployment view of a system, with nodes and their connections being the primary components. The diagram is used to determine the software deployment strategy on the hardware and how the design-created software architecture is connected to the actual system architecture, where the software will run as a node. Communication channels are utilized to depict the links, as there may be numerous nodes involved.

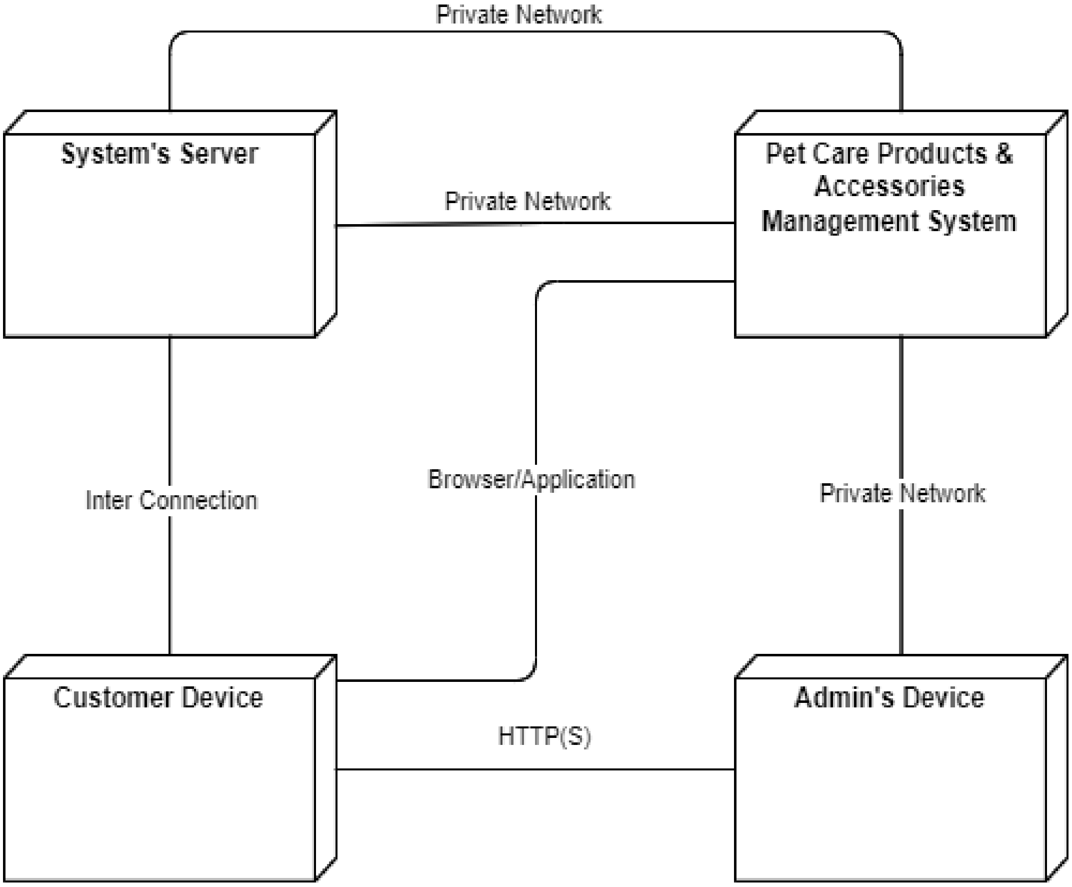


Figure 4.2.8.1: Deployment diagram for the proposed system

4.2.9 Collaboration Diagram

The cooperation diagram is a UML diagram that shows how objects in a system interact with one another. Unlike the sequence diagram, it doesn't focus on message flow, but rather on the relationships between objects in the system. The cooperation diagram depicts the object architecture of the system based on object-oriented programming principles. Objects in the system possess a variety of characteristics, and they are linked to each other. The collaboration diagram, also known as a communication diagram, is another name for the cooperation diagram and is used to visualize the object architecture within the system.

.

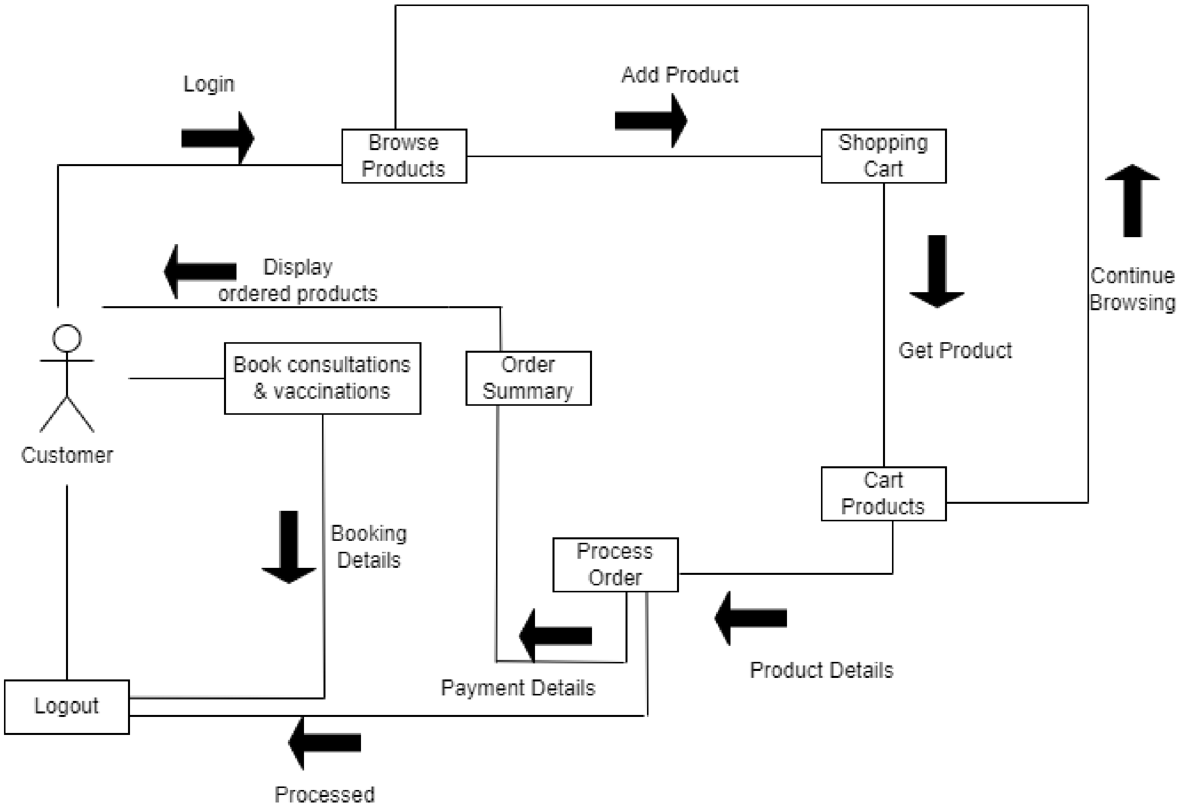


Figure 4.2.9.1: Collaboration diagram for the proposed system

4.3 USER INTERFACE DESIGN USING FIGMA

Fig 4.3.1 Registration Form

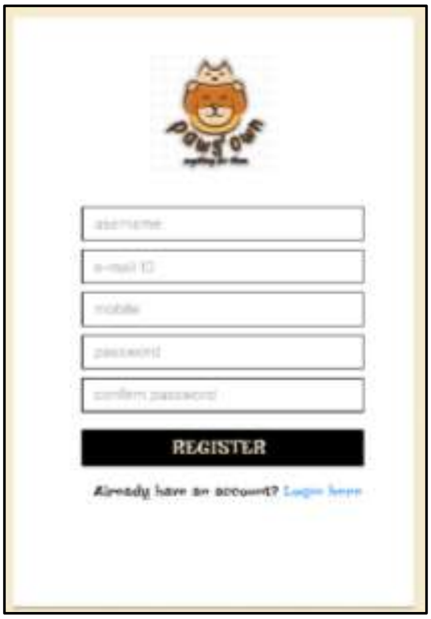


Fig 4.3.2 Login Form

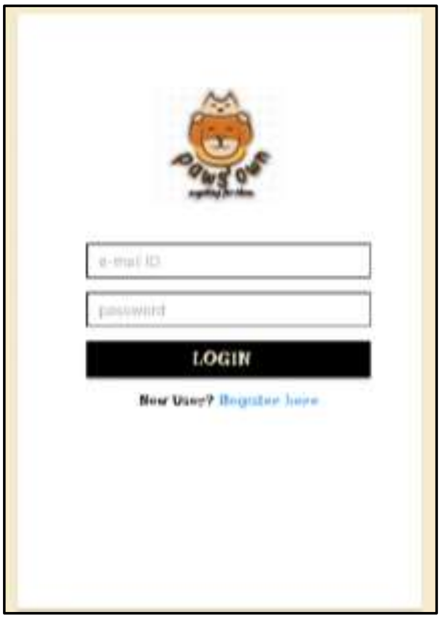


Fig 4.3.3 Consultation Form

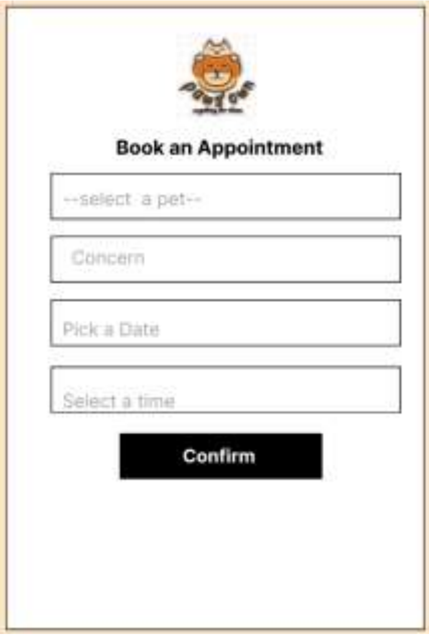


Fig 4.3.4 Vaccination Form

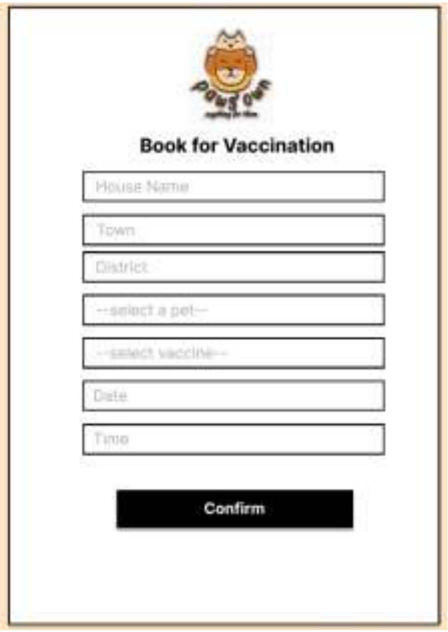


Fig 4.3.5 Customer Home Page

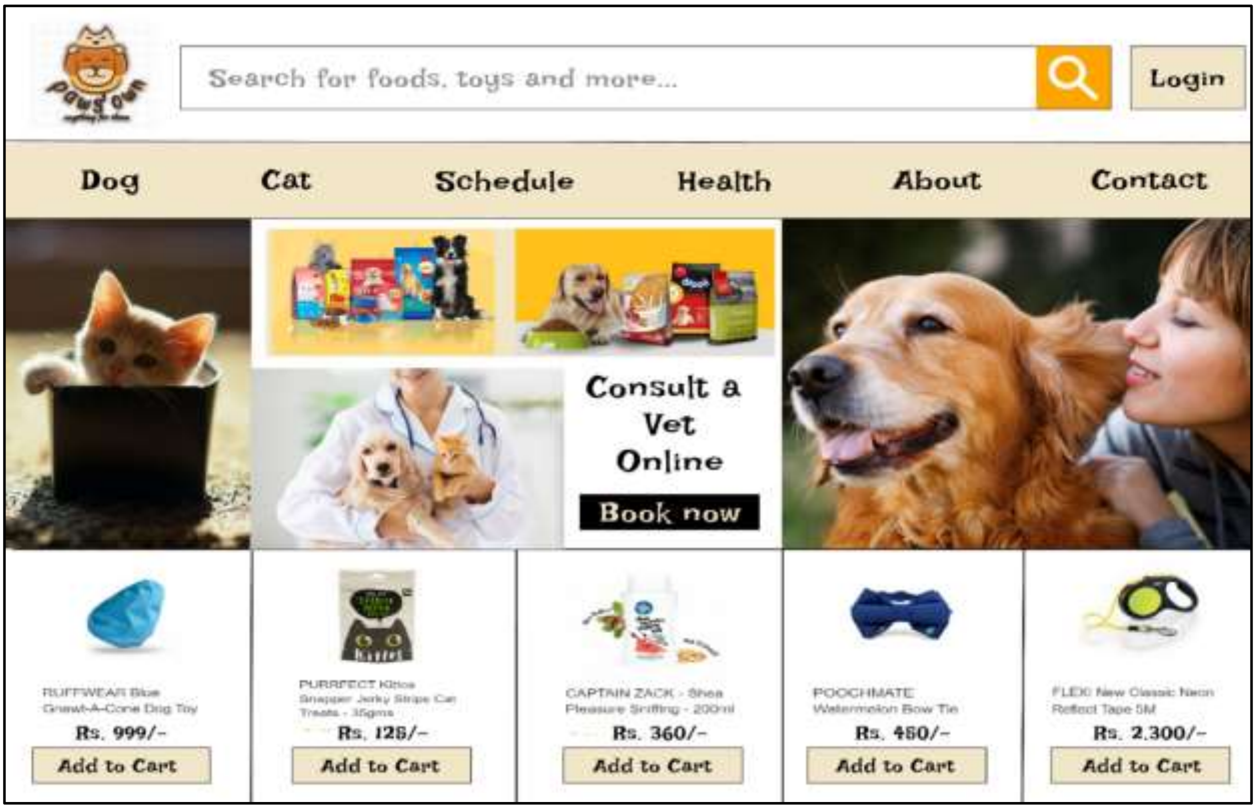
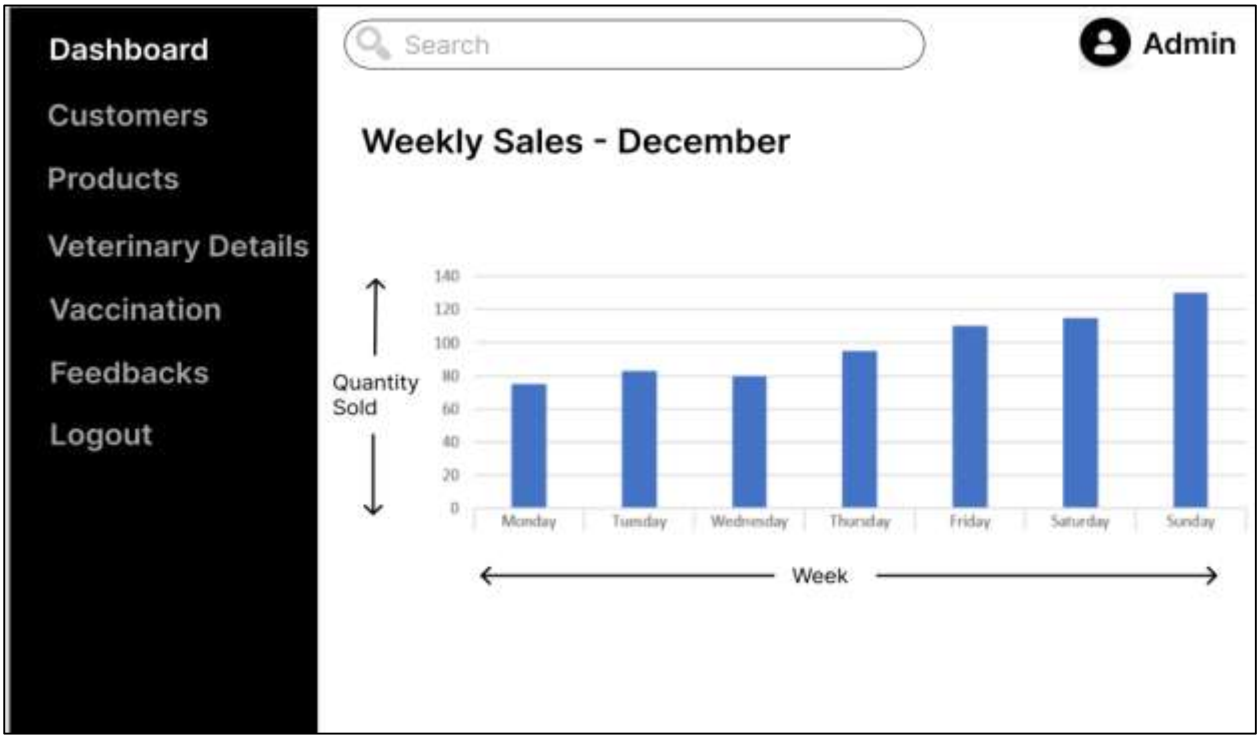

Pawsome
Search for foods, toys and more...
Login
Dog
Cat
Schedule
Health
About
Contact
Consult a Vet Online
Book now
RUFFWEAR Blue Gnawt-A-Cone Dog Toy
Rs. 999/-
Add to Cart
PURRFECT Kitos Snapper Jerky Strips Cat Treats - 35gms
Rs. 128/-
Add to Cart
CAPTAIN ZACK - Shea Pleasure Sniffing - 200ml
Rs. 360/-
Add to Cart
POOCHMATE Watermelon Bow Tie
Rs. 480/-
Add to Cart
FLEXI New Classic Neon Reflect Tape 5M
Rs. 2,300/-
Add to Cart

Fig 4.3.6 Sales Prediction



4.4 DATABASE DESIGN

A database is a structured system that stores information and facilitates efficient retrieval of data. The primary objective of any database is to safeguard the data it stores. The database design process comprises two stages: information level design, which is independent of any specific DBMSs, involves determining user requirements and designing a database that meets those needs as accurately as possible. The second stage, Physical Level Design, involves transforming the information level design into a design that is specific to the DBMS that will be used to build the system. This stage focuses on the features of the specific DBMS to be utilized. Data Integrity and Data Independence are the primary goals of data organization within a database.

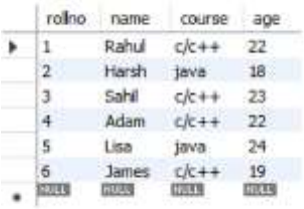
4.4.1 Relational Database Management System (RDBMS)

The relational model is a way to represent a database as a collection of relationships, with each relationship being represented as a table of values or a file of records. In the formal language of the relational model, each row is referred to as a tuple, the column header is known as an attribute, and the table itself is called a relation. The tables in a relational database are grouped and named differently. Each row in the table corresponds to a group of related values, similar to a story.

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. 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. Two different kinds of keys are primary keys and foreign keys. A primary key 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. Redundancy is eliminated through normalization, which in turn reduces the burden on the computer's resources. This is achieved by removing redundant groups of data.

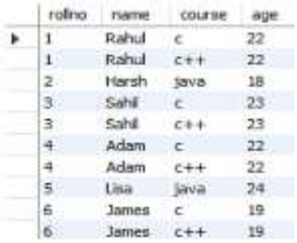
First Normal Form: The First Normal Form (1NF) dictates that all attributes in a table should be atomic, without any sub-relations or nested attributes. To adhere to 1NF, data must be separated into tables with similar data types and each table should have a Primary Key or Foreign Key. This separation helps remove duplicate groups of data and creates new relations for non-atomic attributes or nested relations. A table is said to be in 1NF only if it follows the rule of having unique and non-repeating primary key values.



rollno	name	course	age
1	Rahul	c/c++	22
2	Harsh	java	18
3	Sahil	c/c++	23
4	Adam	c/c++	22
5	Lisa	java	24
6	James	c/c++	19
NULL	NULL	NULL	NULL

Fig 4.4.2.1 Table not in 1NF

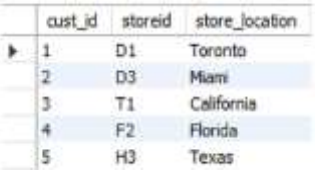
The table shown above displays a violation of the First Normal Form since the "course" column contains two values. To conform to the First Normal Form, it is necessary to split this column.



rollno	name	course	age
1	Rahul	c	22
1	Rahul	c++	22
2	Harsh	java	18
3	Sahil	c	23
3	Sahil	c++	23
4	Adam	c	22
4	Adam	c++	22
5	Lisa	java	24
6	James	c	19
6	James	c++	19

Fig 4.4.2.2 Table in 1 NF

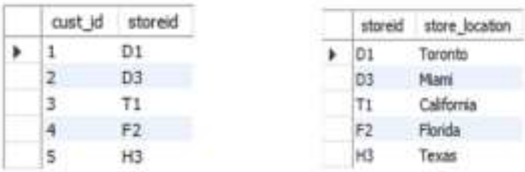
Second Normal Form: To meet the requirements of the Second Normal Form, it is necessary that no non-key attribute depends on only a portion of the primary key if the primary key comprises multiple attributes. To qualify for the second normal form, a relation must meet the primary key requirements of the first normal form and ensure that all non-primary key attributes are entirely dependent on the primary key.



cust_id	storeid	store_location
1	D1	Toronto
2	D3	Miami
3	T1	California
4	F2	Florida
5	H3	Texas

Fig 4.4.2.3 Table not in 2NF

This table does not meet the criteria for second normal form because the store location is dependent on only a part of the primary key, which is "storeID." Therefore, to satisfy the second normal form. the table needs to be split.

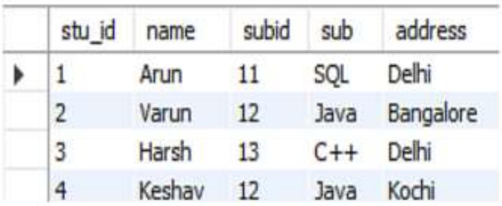


cust_id	storeid
1	D1
2	D3
3	T1
4	F2
5	H3

storeid	store_location
D1	Toronto
D3	Miami
T1	California
F2	Florida
H3	Texas

Fig 4.4.2.4 Table in 2NF

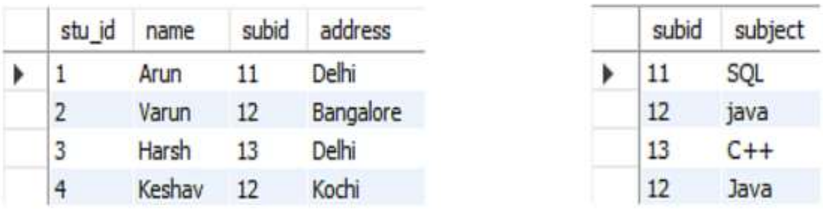
Third Normal Form: To meet the Third Normal Form, a table should not have a non-key attribute that is determined by a combination of other non-key attributes. Therefore, a relation must satisfy the Second Normal Form, and the non-key attributes must be fully dependent on the primary key. Here's an illustration of a table containing columns for student ID, student name, subject ID, subject name, and student address, which exemplifies this requirement.



	stu_id	name	subid	sub	address
▶	1	Arun	11	SQL	Delhi
	2	Varun	12	Java	Bangalore
	3	Harsh	13	C++	Delhi
	4	Keshav	12	Java	Kochi

Fig 4.4.2.5 Table not in 3NF

The table of students given above violates the third normal form because stu_id determines sub via sub_id, creating a transitive functional dependency. To conform to the requirements of the third normal form, we need to eliminate the transitive functional dependency by splitting the table.



	stu_id	name	subid	address
▶	1	Arun	11	Delhi
	2	Varun	12	Bangalore
	3	Harsh	13	Delhi
	4	Keshav	12	Kochi

	subid	subject
▶	11	SQL
	12	java
	13	C++
	12	Java

Fig 4.4.2.6 Table in 3NF

4.4.3 Sanitization

Sanitizing To sanitize data means to eliminate any disallowed elements. One of the most common tasks in web development is sanitizing user input. PHP has a built-in filter extension that facilitates the process of sanitizing data, including email addresses, URLs, IP addresses, and so on. The PHP filter extension allows you to sanitize and validate external input by using various functions. Its purpose is to make the process of sanitizing data simpler and faster. For instance, by using a flag in the example, this method ensures that all characters except letters, numbers, and the following characters #$%&'*+-=?_`{|}~@.[] are removed from the code. Sanitizing inputs can be a useful option when the input format is flexible but relatively predictable, such as with phone numbers or other free-text fields. You can sanitize inputs in various ways, such as using a whitelist, a blacklist, or escaping input.

4.4.4 Indexing

Indexing is a method to enhance a database's performance by minimizing the number of disk accesses needed during query processing. An index is a specific data structure that enables rapid access to a database table's data. There are two types of indexing: primary indexing and secondary indexing. A primary index is an ordered file with two fields: the primary key and a pointer to the data block. In a primary index, entries in the index table are always associated with each other. A non-clustering index, also called a secondary index in DBMS, is created based on a field with a unique value for each record, which can be a candidate key. The secondary index is generated by a field with a unique value for each record, and it should be a candidate key. It is also called a non-clustering index. This two-level indexing technique is used to reduce the first level's mapping size.

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**

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

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

The process of software testing involves evaluating and confirming that a software application or product performs as intended, which can prevent bugs, reduce development costs, and improve performance. Validation, which examines whether a product complies with relevant specifications, is often used in conjunction with software testing. Verification, including reviews, analyses, inspections, and walkthroughs, is a type of software testing that helps to ensure that what has been specified matches what the user desired. Static analysis and dynamic analysis are two techniques that are utilized in software testing. Static analysis involves inspecting the source code to find potential issues and collect statistical data, whereas dynamic analysis focuses on how the software functions during execution to gather information about timing profiles, execution traces, and test coverage. The process of software testing is a well-planned and systematic set of activities that begins with individual modules and gradually progresses towards system integration. To establish testing objectives, various guidelines can be followed, and to ensure effective system testing, the testing process must be carried out efficiently. They are:

- Executing a program with the aim of detecting an error is referred to as testing.
- A test case that has a high chance of discovering an error that has not yet been detected is considered to be a good test case.
- A test is deemed successful if it uncovers an error that was not previously detected.

Testing software according to the objectives mentioned above can aid in identifying software defects. Software programs can be tested in three ways, which include checking for correctness, implementation effectiveness, and computational power. Testing for correctness aims to ensure that the program performs exactly as intended. However, this can be challenging, particularly for large programs.

5.1 TEST PLAN

A test plan is a document that outlines the steps required to conduct various testing methods. It specifies the activities to be performed during the testing process. Typically, software developers create a computer program, related data structures, and supporting documentation. In order to address concerns with developers evaluating their own work, there is an independent test group (ITG). The test strategy should incorporate information about mean time to failure, the cost of identifying and correcting problems, remaining defect frequency or density, and testing hours for regression testing. The levels of testing may 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

Unit testing is a method of software testing that involves evaluating individual units or components, such as computer program modules, operating procedures, and usage procedures, to assess their suitability for use. This testing technique involves examining each module by the developer to identify issues and evaluate its performance. Unit testing is typically performed on the software components during application development, which can include specific functions or techniques. The first level of testing in the software development life cycle or V Model is usually performed by the developer, and it involves unit testing. This testing technique is frequently used by developers.

5.2.2 Integration Testing

Integration testing is a technique of software testing that involves testing several software components together as a single unit. In most software projects, there are multiple modules created by different developers. Integration testing is performed to identify issues that may occur when different software modules are integrated, ensuring that the entire software program works as intended. This testing can be challenging since it may be difficult to locate errors due to the complex nature of the entire program. Additionally, this process may be iterative since new errors may arise after resolving previous ones. Integration testing also helps to eliminate different program structure variations and establishes a unique program structure.

5.2.3 Validation Testing or System Testing

Validation testing is a method of evaluating software at different stages of development or after completion to verify that it meets the specified business requirements. This testing ensures that the end product meets the client's needs and is appropriate for its intended use when it is deployed in the appropriate environment. The main question that validation testing answers is whether the right product is being built. It is a critical testing process in software engineering that verifies if the existing system conforms to the system requirements and performs its intended functions, while meeting the goals and needs of the organization. Validation testing is vital for software testers who strive to be among the best in the field. After the validation testing stage, the secondary process is software verification and validation testing.

5.2.4 Output Testing or User Acceptance Testing

User Acceptance Testing (UAT) is a software testing phase where the end-users or target user group test the software in real-life scenarios. It is usually the final testing stage before the product is released to the market. The primary objective of UAT is to confirm that the software functions as intended and can handle practical tasks in accordance with the requirements. UAT allows users to interact with the software and provide feedback on any missed features or issues before the software is officially launched. The feedback from early testers is used by developers to make necessary adjustments, ensuring quality and transparency, and saving time and software costs. Successful UAT can validate business requirements and provide access to actual instances and data.

5.2.5 Automation Testing

Automated testing is a software testing approach that utilizes specialized tools to execute a set of predefined test cases. Unlike manual testing, which requires a person to perform test steps manually, automated testing software can perform tasks such as inputting test data into the system, comparing actual results with expected ones, and generating detailed test reports. While implementing automated testing requires a significant investment of resources and funds, its purpose is not to entirely eliminate manual testing. Instead, the objective is to decrease the number of test cases that need to be executed manually.

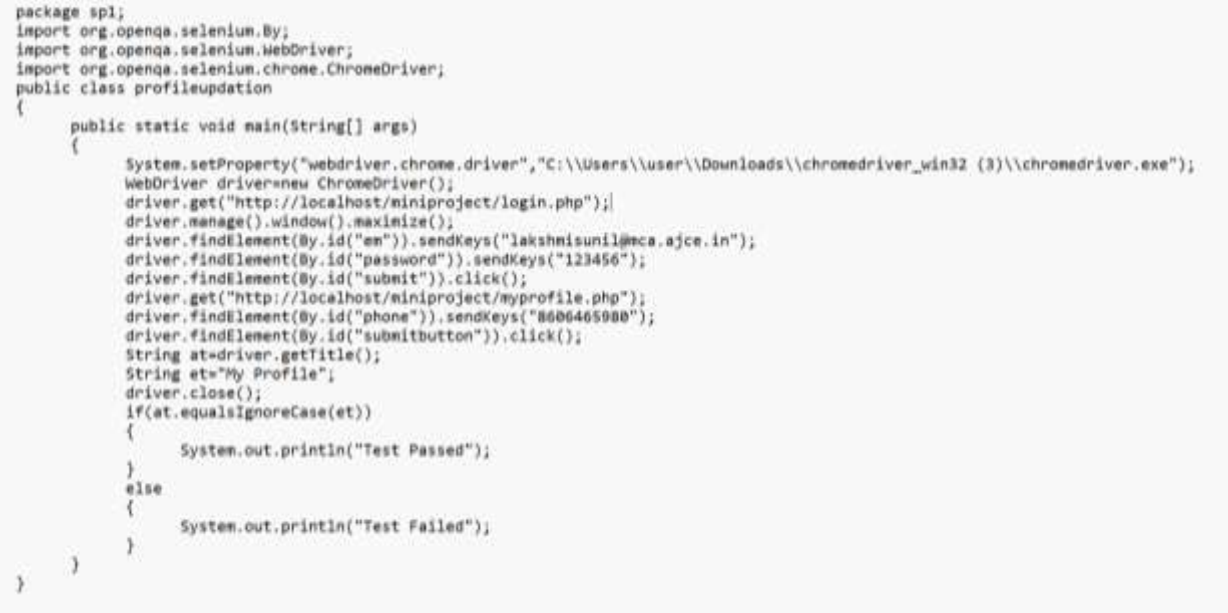
5.2.6 Selenium Testing

Developers should be familiar with Selenium for effective web application development. Selenium is an open-source tool for automated testing, which includes a collection of tools with specific purposes. Selenium is capable of performing various types of testing, such as system testing, end-to-end testing, compatibility testing, regression testing, integration testing, and performance testing. Selenium is not just a single tool, but a combination of four components that cater to diverse testing needs of different organizations. The four components of Selenium are as follows:

- **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.

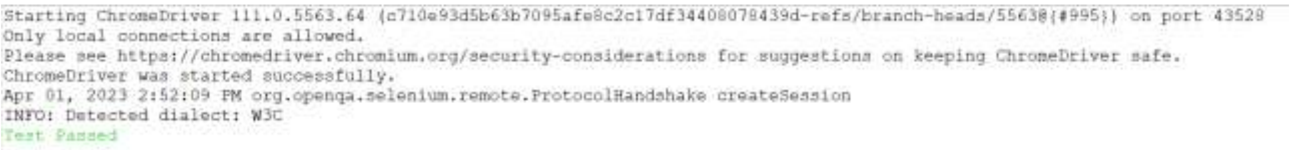
Test Case 1- Profile Updation

Code



```
package spl;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class profileupdation
{
	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/myprofile.php");
		driver.findElement(By.id("phone")).sendKeys("8606465980");
		driver.findElement(By.id("submitbutton")).click();
		String at=driver.getTitle();
		String et="My Profile";
		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

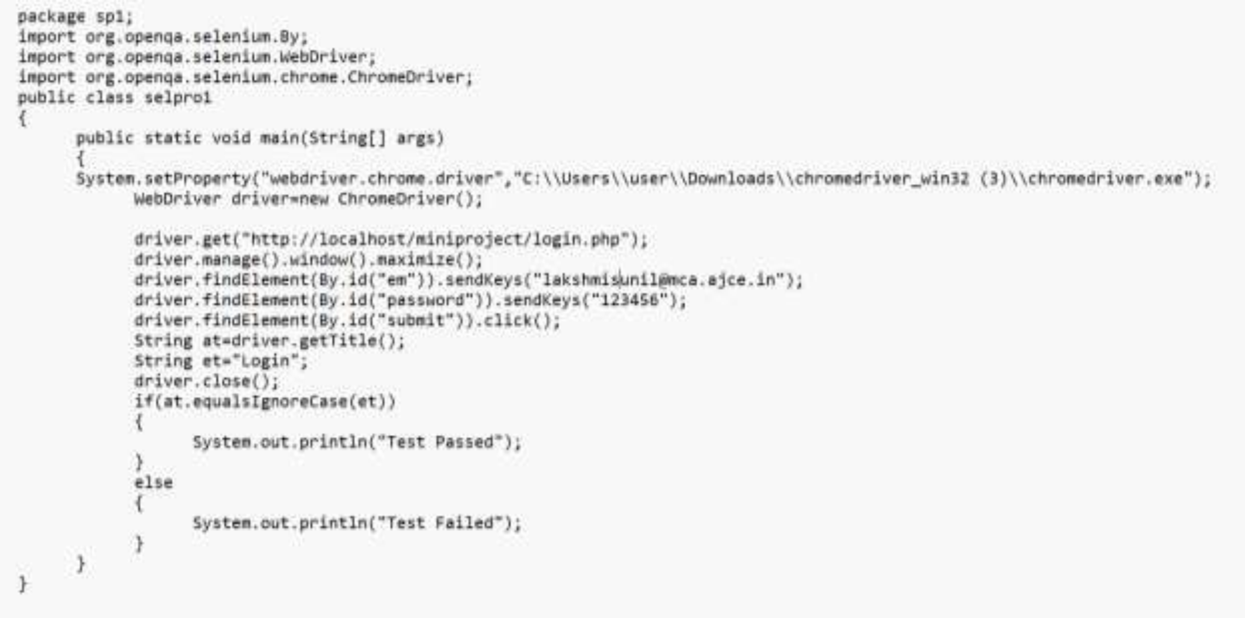
Project Name: PAWS' OWN	
Profile updation Test Case	
Test Case ID: Test 1	**Test Designed By: Lakshmi Sunil**
Test Priority(Low/Medium/High): High	**Test Designed Date: 01/04/2023**
Module Name: Profile updation	**Test Executed By : DR. BIJIMOL T. K.**
Test Title : Update profile with new phone number	**Test Execution Date:01/04/2023**
Description: My Profile updation	
Pre-Condition : New inputs must be valid	

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	No error message	User Logged in	Pass
3	Enter valid password	Password:123456			
4	Click on Login button				
5	Navigate to My Profile Page		My Profile Page	Navigated to My Profile Page	Pass
6	Enter valid phone number	Phone number: 8606465980	No error message	Phone number updated	Pass
7	Click on update button				
8	Enter invalid phone number	Phone number: 099999999	User should not be able to update profile	Error message displayed	Pass
9	Click on Update Profile				

Post-Condition: All inputs are validated and profile is updated successfully.

Test Case 2: Login

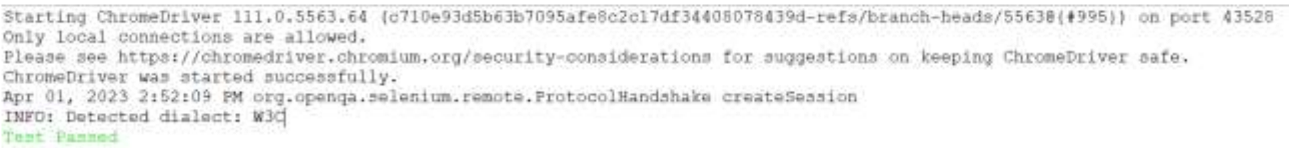
Code



```
package sp1;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class selpro1
{
      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
```

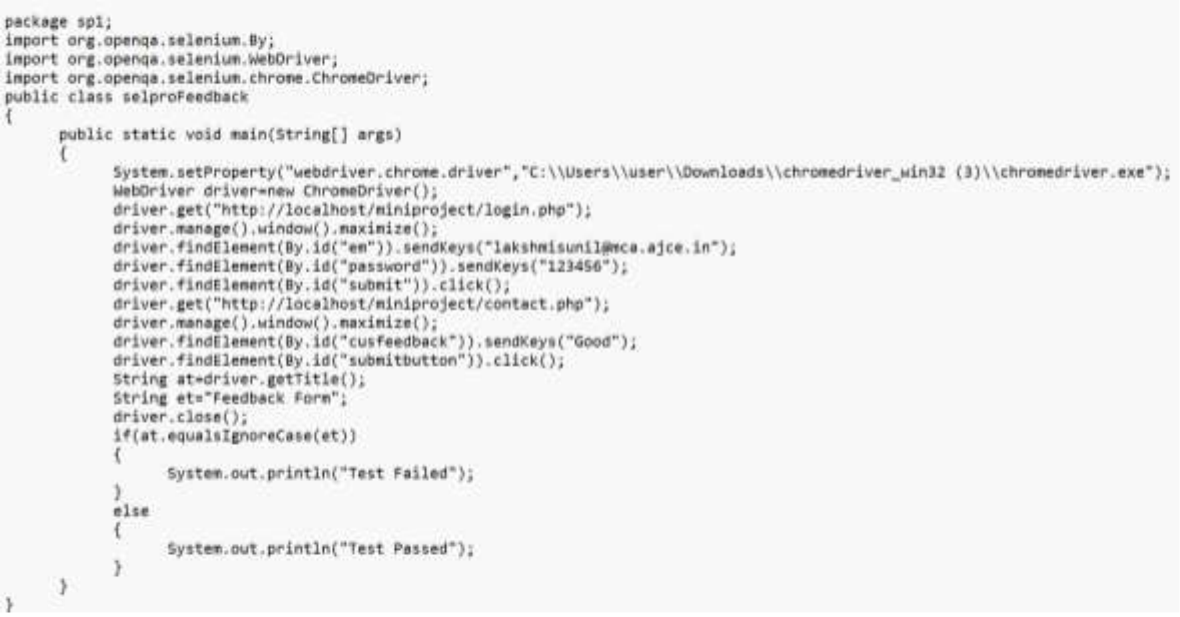
Test report

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					

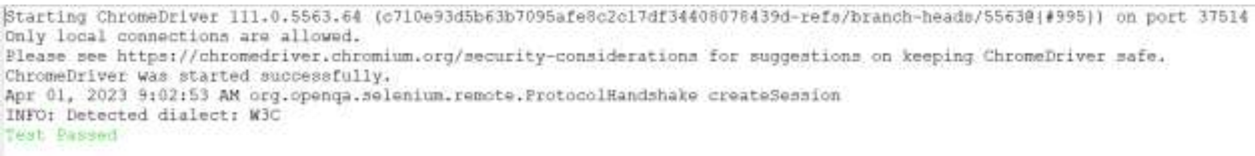
Test Case 3: Submit Feedback

Code



```
package sp1;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
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



```
Starting ChromeDriver 111.0.5563.64 (c710e93d5b63b7095afe8c2c17df34408078439d-refs/branch-heads/5563@{#995}) on port 37514
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 9:02:53 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Test Passed
```

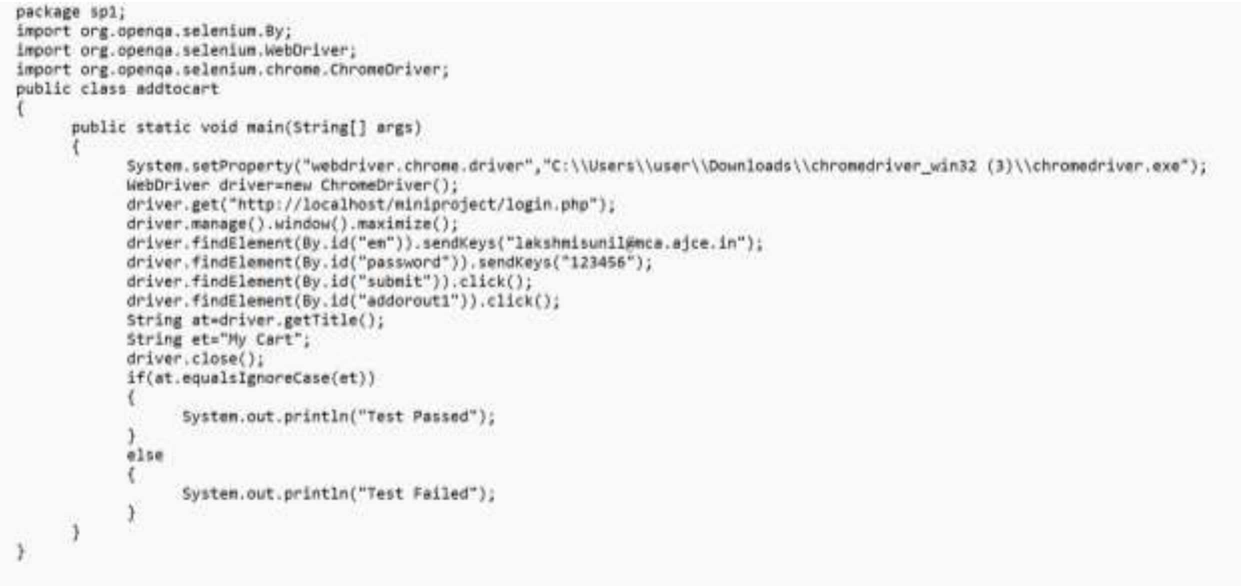
Test Report

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
3	Click on Login button				
4	Enter feedbacks with numbers	Feedback: 8#hufufg	Feedback cannot be submitted	Feedback is not submitted	Pass
5	Click on login button				
Post-Condition: Valid feedback is submitted successfully					

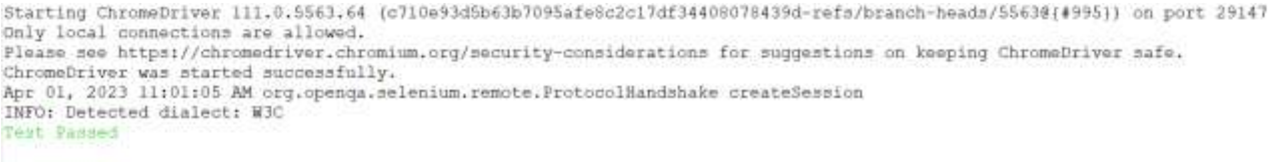
Test Case 4: Add to Cart

Code



```
package sp1;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class addtocart
{
	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.findElement(By.id("addorout1")).click();
		String at=driver.getTitle();
		String et="My Cart";
		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
```

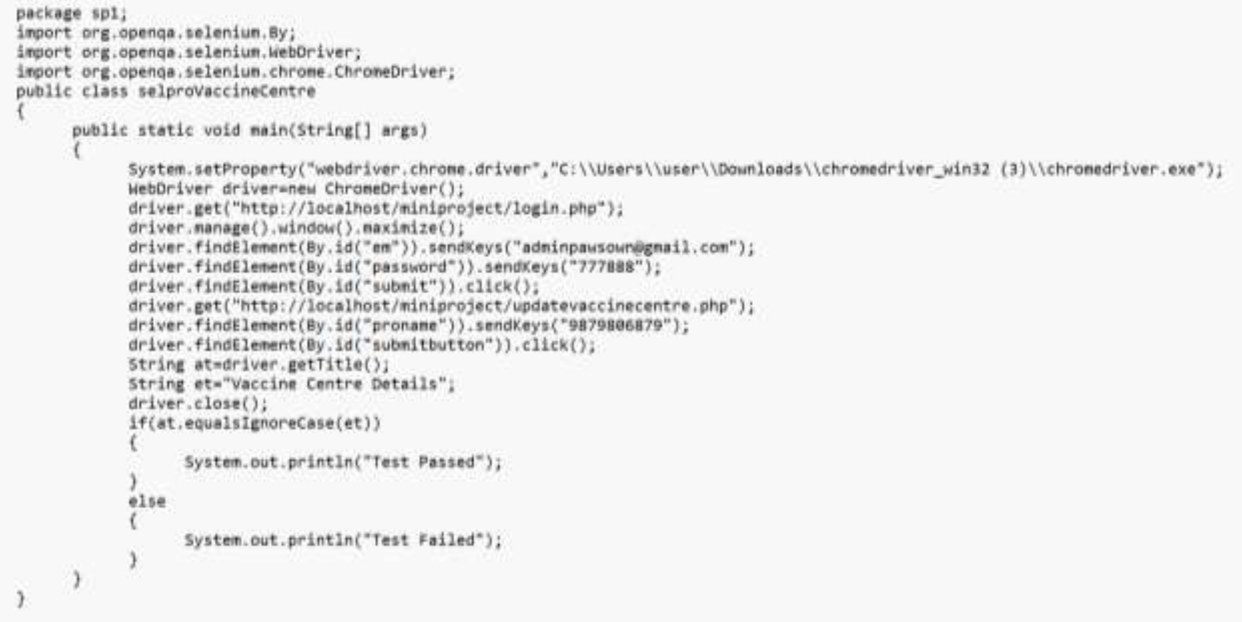
Test Report

<table>
<tr><td colspan="6"><b>Test Case 4</b></td></tr>
<tr><td colspan="6"><b>Project Name: PAWS’ OWN</b></td></tr>
<tr><td colspan="6"><b>Add to Cart Test Case</b></td></tr>
<tr><td colspan="3"><b>Test Case ID:</b> Test_4</td><td colspan="3"><b>Test Designed By: LAKSHMI SUNIL</b></td></tr>
<tr><td colspan="3"><b>Test Priority(Low/Medium/High):</b> High</td><td colspan="3"><b>Test Designed Date: 01/04/2023</b></td></tr>
<tr><td colspan="3"><b>Module Name</b>: Add to Cart</td><td colspan="3"><b>Test Executed By : DR. BIJIMOL T. K.</b></td></tr>
<tr><td colspan="3"><b>Test Title :</b> Add to Cart with productID</td><td colspan="3"><b>Test Execution Date:01/04/2023</b></td></tr>
<tr><td colspan="3"><b>Description:</b> Add to Cart functionality</td><td colspan="3"></td></tr>
<tr><td colspan="6"><b>Pre-Condition :</b> Product with a valid productID can only be added to cart.</td></tr>
<tr><td><b>Step</b></td><td><b>Test Step</b></td><td><b>Test Data</b></td><td><b>Expected Result</b></td><td><b>Actual Result</b></td><td><b>Status(Pass/ Fail)</b></td></tr>
<tr><td>1</td><td>Navigate to Home Page</td><td></td><td rowspan="2">Product should</td><td rowspan="2">Product added</td><td rowspan="2">Pass</td></tr>
<tr><td></td><td>Add to cart of</td><td>“Add to</td></tr>
</table>

2	product with ID=1 is clicked	Cart" clicked	be added to cart	to cart	
3	Navigate to My Cart page				
4	Navigate to home page		Product should not be added to cart	Product is not added to cart	Pass
5	Add to cart of product with no ID is clicked	"Add to Cart" clicked			
6	No navigation to My Cart page				
Post-Condition: All inputs are validated and product added to cart successfully.					

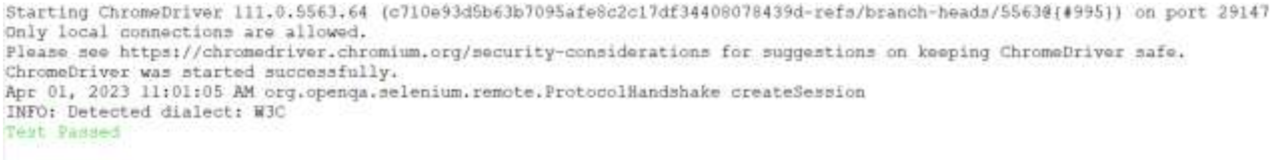
Test Case 5: Edit Vaccine Centre Phone

Code



```
package sp1;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class selproVaccineCentre
{
	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("adminpawsown@gmail.com");
		driver.findElement(By.id("password")).sendKeys("777888");
		driver.findElement(By.id("submit")).click();
		driver.get("http://localhost/miniproject/updatevaccinecentre.php");
		driver.findElement(By.id("proname")).sendKeys("9879806879");
		driver.findElement(By.id("submitbutton")).click();
		String at=driver.getTitle();
		String et="Vaccine Centre Details";
		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 5	
Project Name: PAWS' OWN	
Edit Vaccine Centre Phone Test Case	
Test Case ID: Test_5	**Test Designed By: LAKSHMI SUNIL**
Test Priority(Low/Medium/High): High	**Test Designed Date: 01/04/2023**
Module Name: Edit Phone	**Test Executed By : DR. BIJIMOL T. K.**
Test Title : Edit Phone number of Vaccine Centre	**Test Execution Date:01/04/2023**
Description: Update Functionality	
Pre-Condition : Phone number should be altered with valid input.	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to Vaccine Centre details page		Navigate to Vaccine Centre details page	Navigated to vaccine centre details page	Pass
2	Click on Edit	Edit clicked	Phone number should be updated	Phone number updated	Pass
3	Enter valid input	Phone: 9879806879			
4	Click "Save"	Phone number updated			
5	Enter invalid phone number or NULL	Phone: "- or Phone : NULL	Phone number cannot be updated	Phone number not upadted	Pass
6	Click "Save"	"Add to Cart" clicked			

Post-Condition: Input is validated and phone number updated successfully.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

To bring the goals and plans of a project into effect, the project implementation (or project execution) phase is employed. During this stage of the project cycle, planned benefits are actualized, making it the most critical stage. The primary objective of implementation is to put the action plan into operation, manage resources effectively, and monitor progress. Poorly planned or managed implementation may result in chaos and disarray. Implementation includes all of the necessary steps to transition from the old system to the new system.

6.2 IMPLEMENTATION PROCEDURES

Implementation refers to the act of putting a plan, design, policy, or any other idea into action. This means that action needs to follow any prior thinking in order for something to actually happen. To access the results of the system, the user must first ensure that the server programs are up and running on the server.

6.2.1 User Training

The provision of guidelines on device usage, system error analysis, and the differentiation between device-related issues and software-related problems constitute user training. In the development of computer-based information systems, end-user training is vital to enable employees to resolve their own issues. The primary focus of user training is the functionality of the technology.

6.2.2 Training on Application Software

Before learning how to use the new application software, the user must have basic computer literacy training. Then, the user should be taught about the fundamental principles of the new system such as the working of screens, potential errors that can occur while entering data, how each entry is verified, and how to modify entered dates.

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.

6.2.4 Hosting

Web hosting refers to the service of providing storage space and access for websites on the internet. In order for a website to be accessible on the internet, it must be stored on a server that is connected to the internet 24/7. A web hosting company provides this service, allowing individuals and businesses to store their website files, data, and applications on a server that is maintained and managed by the hosting company.

AWS (Amazon Web Services)

Amazon Web Services, is a cloud computing platform offered by Amazon. AWS offers a range of services that are well-suited for hosting a website. One of the most commonly used services for website hosting on AWS is Amazon Elastic Compute Cloud (EC2). EC2 provides scalable compute capacity in the cloud and allows users to launch and manage virtual machines on the cloud. Procedure to host a website in AWS is as follows:

Step 1: Sign up for an AWS account and create an IAM user with appropriate permissions.

Step 2: Launch an Amazon EC2 instance and choose an appropriate operating system.

Step 3: Connect to the EC2 instance using SSH and install the web server, Apache.

Step 4: Upload your website files to the EC2 instance and configure the web server.

Step 5: Configure the web server to serve the website

Step 6: Set up a domain name and point it to the IP address of the EC2 instance

Hosted Website Link: https://pawsownindia.ddns.net/pawsown/

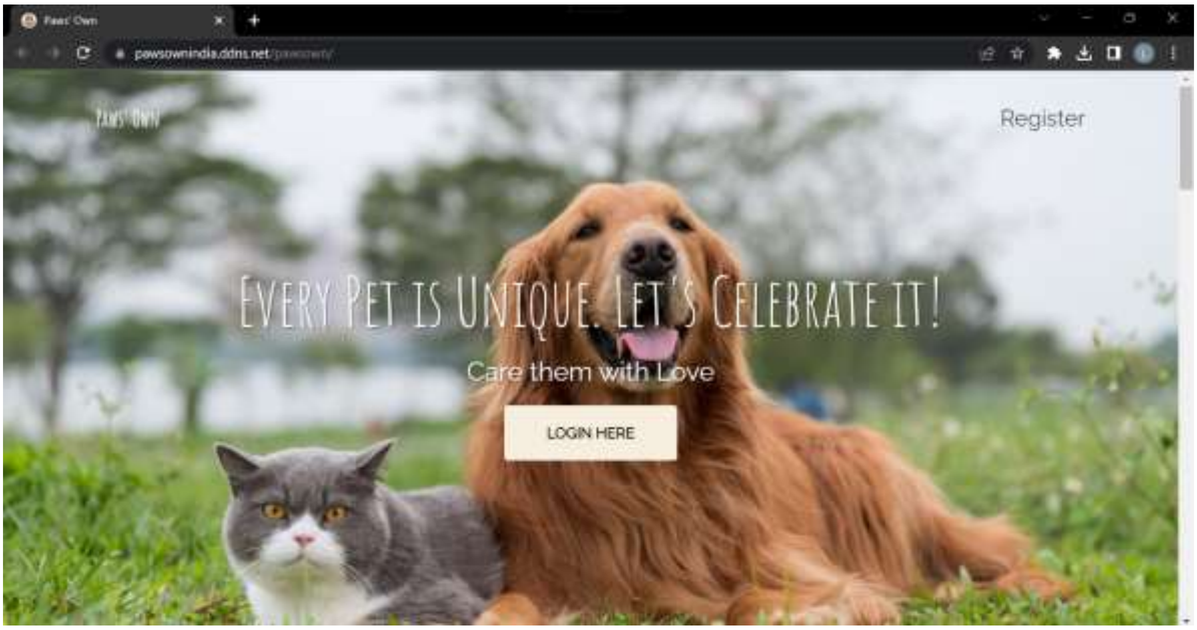

Paws' Own
pawsownindia.ddns.net/pawsown/
PAWS' OWN
Register
EVERY PET IS UNIQUE. LET'S CELEBRATE IT!
Care them with Love
LOGIN HERE

Fig 6.2.4.1 Hosted Website

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. PAWS' OWN, a web-based application that provides pet care items and services, offers a user-friendly system that is highly advantageous to customers. The proposed solution aims to address all of the current system's drawbacks and operational challenges. With the new system, pet owners will be able to book appointments with veterinarians to address their pets' various health concerns, thereby reducing manual labor. Additionally, it provides the option for users to register and log in, search for products, add items to their carts, and make purchases. The suggested system aims to reduce user workload and minimize mental stress, making it more time and cost-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

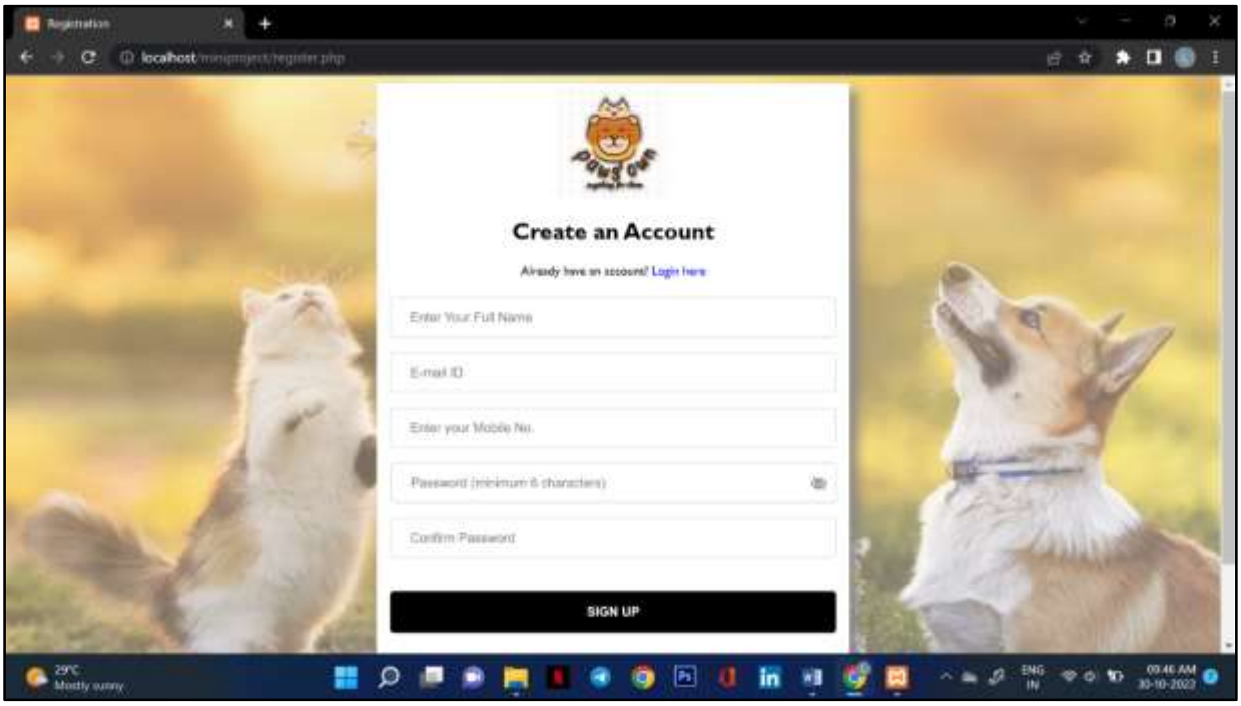
9.1 Sample Code

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



Create an Account

Already have an account? Login here

Enter Your Full Name

E-mail ID

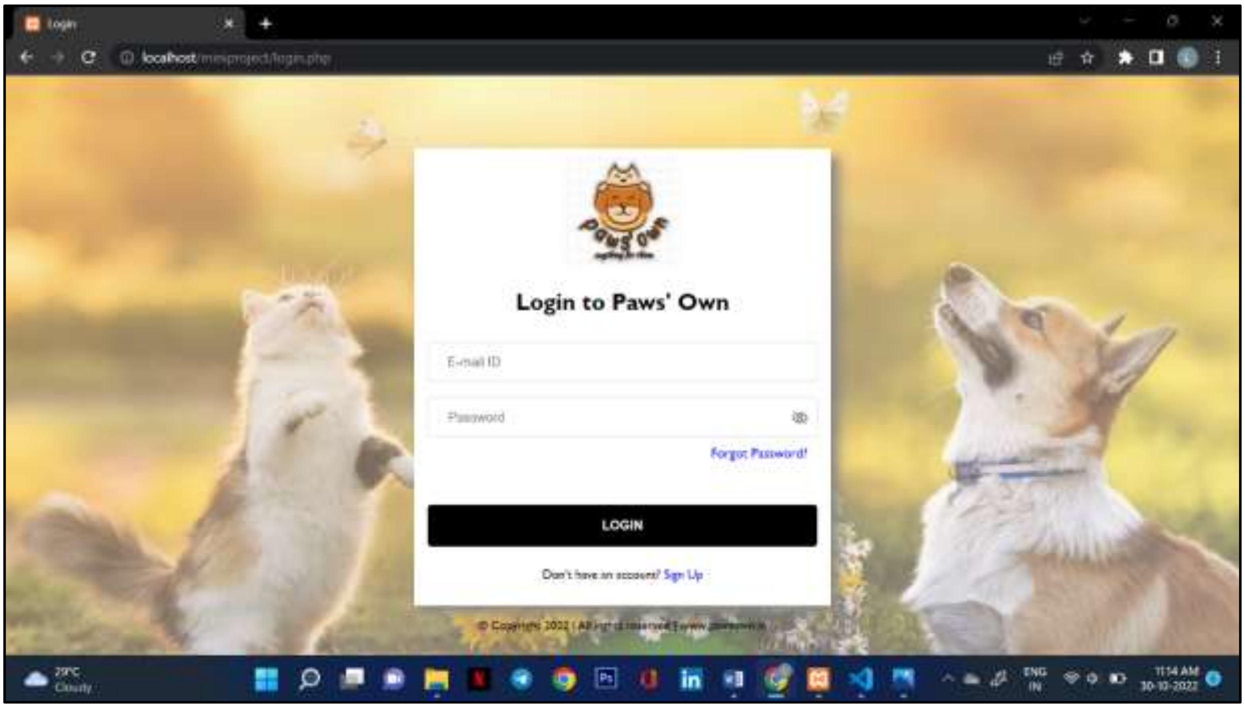
Enter your Mobile No.

Password (minimum 6 characters)

Confirm Password

SIGN UP

1.2.2 Login



Login to Paws' Own

E-mail ID

Password

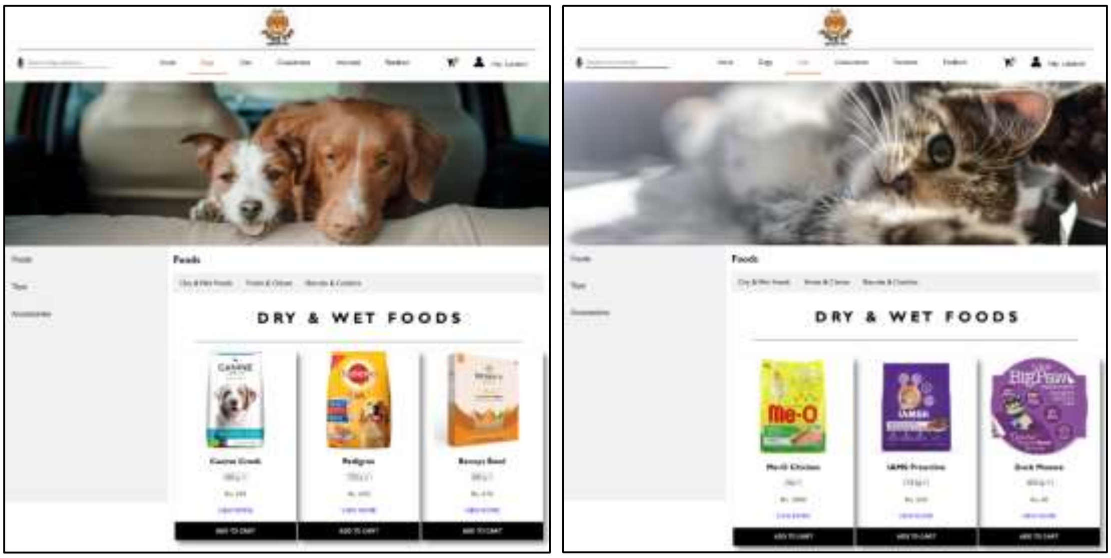
Forgot Password!

LOGIN

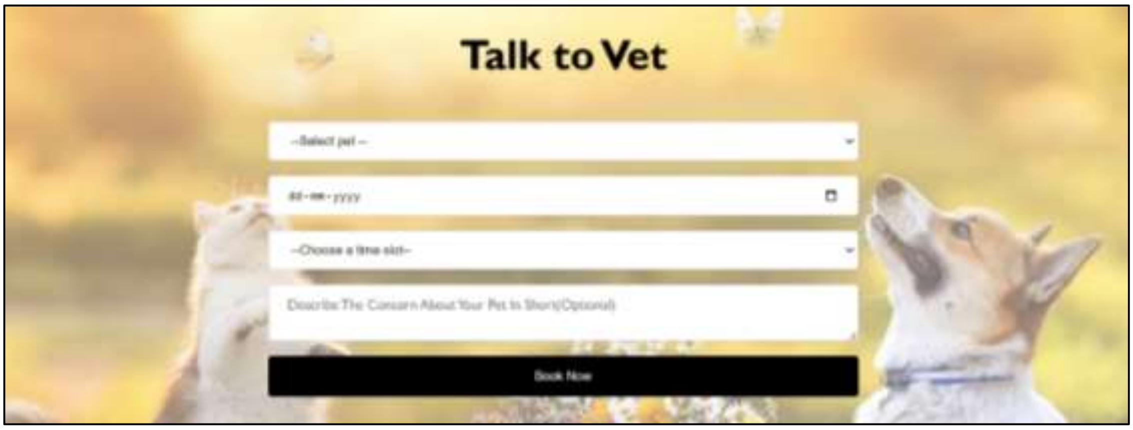
Don't have an account? Sign Up

© Copyright 2022 | All rights reserved | www.pawsown.in

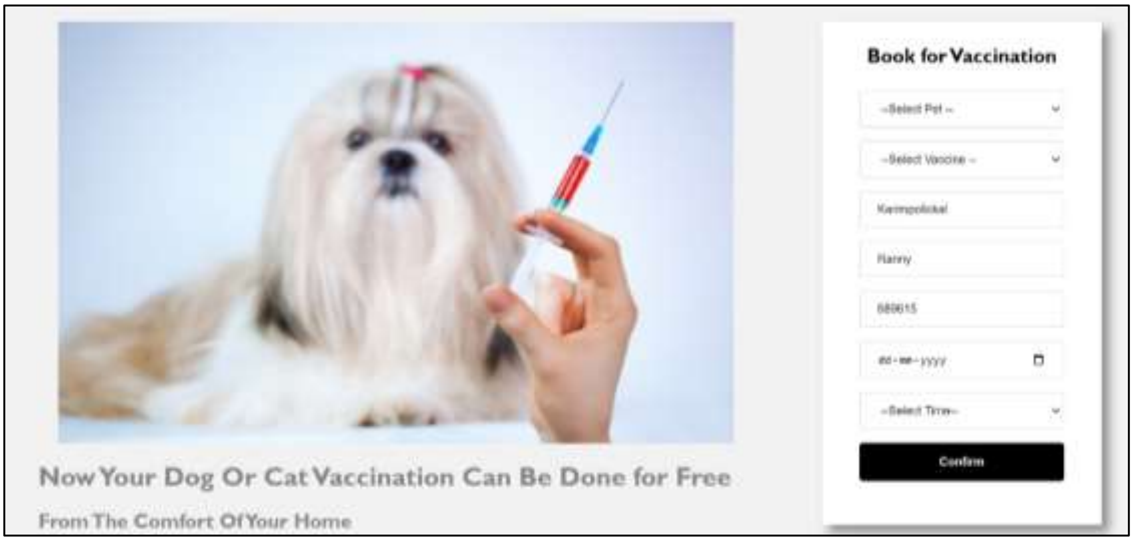
1.2.3 Customer : Home Page – Dogs & Cats



1.2.4 Customer : Appointment Form



1.2.5 Customer : Vaccination Booking Form

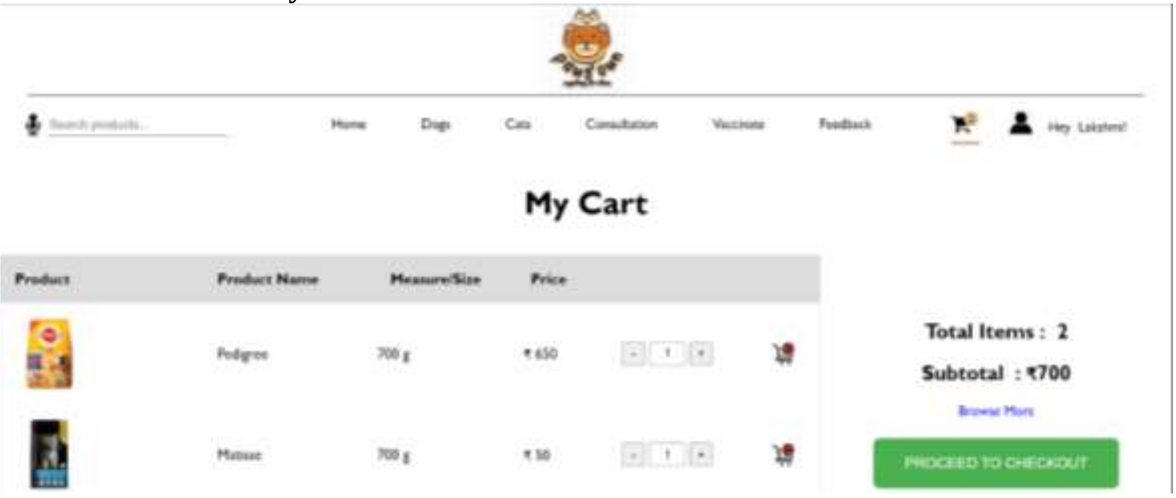


Karimpolickal

Hanny

686615

1.2.6 Customer : My Cart



Search products...

Home Dogs Cats Consultation Vaccinate Feedback Hey Lakshmi!

My Cart

Product	Product Name	Measure/Size	Price		
	Pedigree	700 g	₹ 650	- 1 +	
	Matisse	700 g	₹ 50	- 1 +	

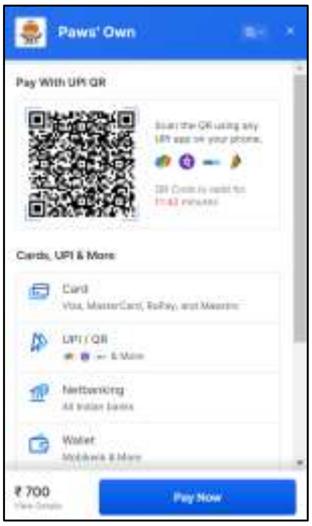
Total Items : 2

Subtotal : ₹700

Browse More

PROCEED TO CHECKOUT

1.2.7 Customers : Payment



Paws' Own

Pay With UPI QR

Scan the QR using any UPI app on your phone.

QR Code is valid for 11:43 minutes

Cards, UPI & More

Card
Visa, MasterCard, RuPay, and Maestro

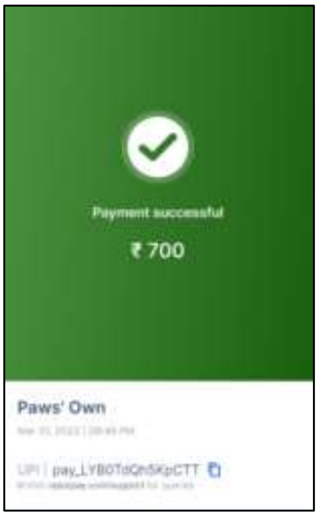
UPI / QR
& More

Netbanking
All Indian banks

Wallet
Mobikwik & More

₹ 700
View Details

Pay Now



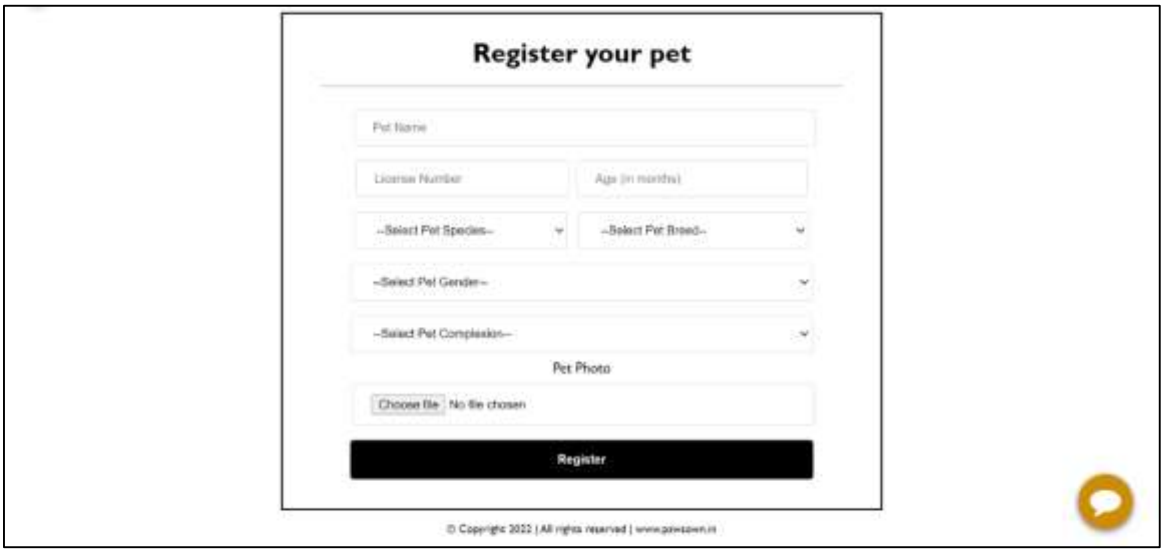
Payment successful

₹ 700

Paws' Own

UPI | pay_LYBOTdQn5KpCTT

1.2.8 Customer : Pet Registration Form



Register your pet

Pet Name

License Number

Age (in months)

--Select Pet Species--

--Select Pet Breed--

--Select Pet Gender--

--Select Pet Complexion--

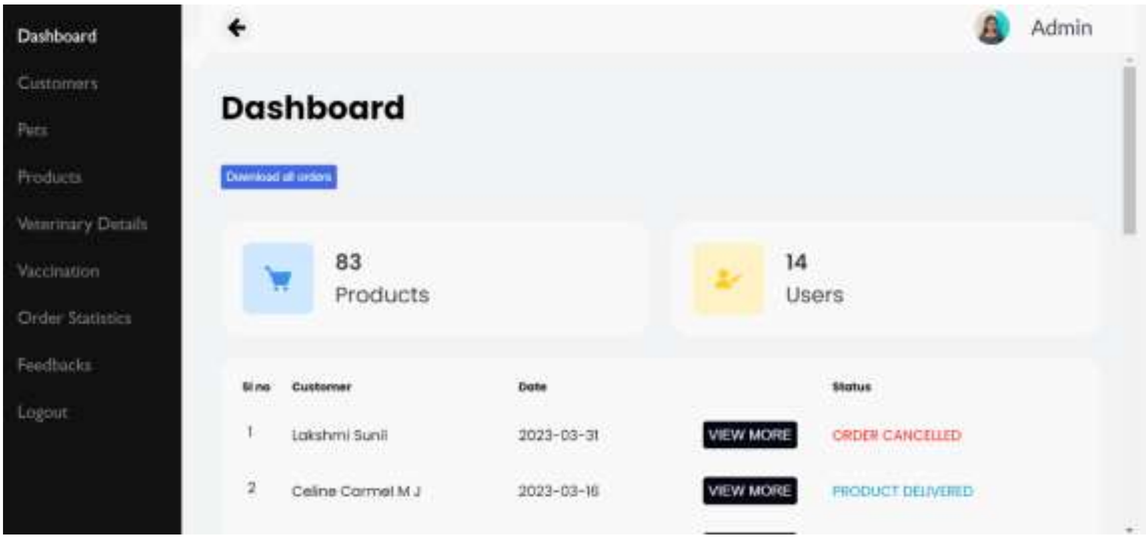
Pet Photo

Choose file No file chosen

Register

© Copyright 2022 | All rights reserved | www.pawsown.in

1.2.9 Admin – Dashboard



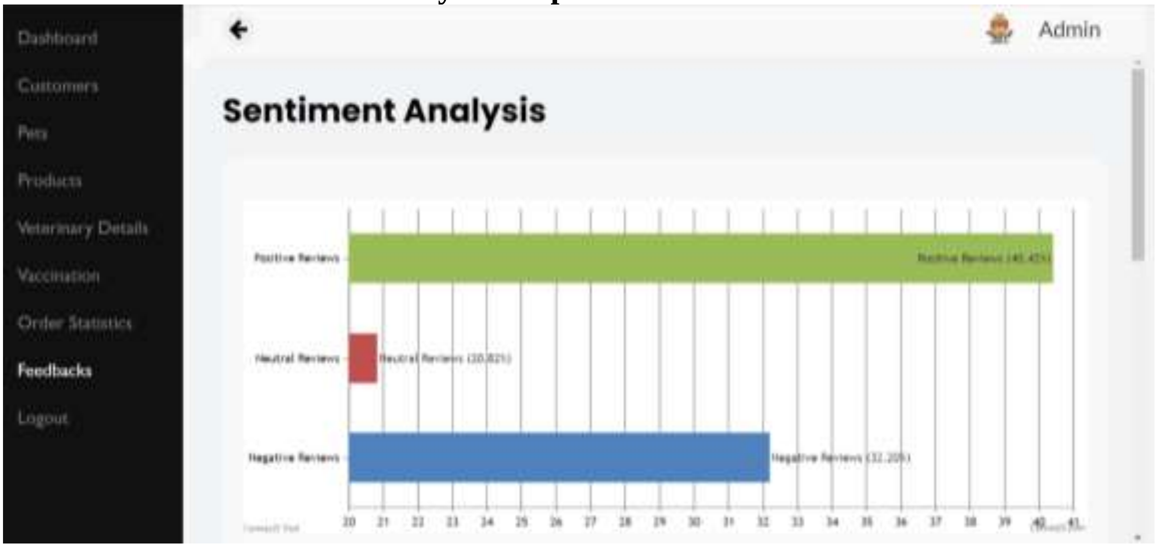
Dashboard

83 Products

14 Users

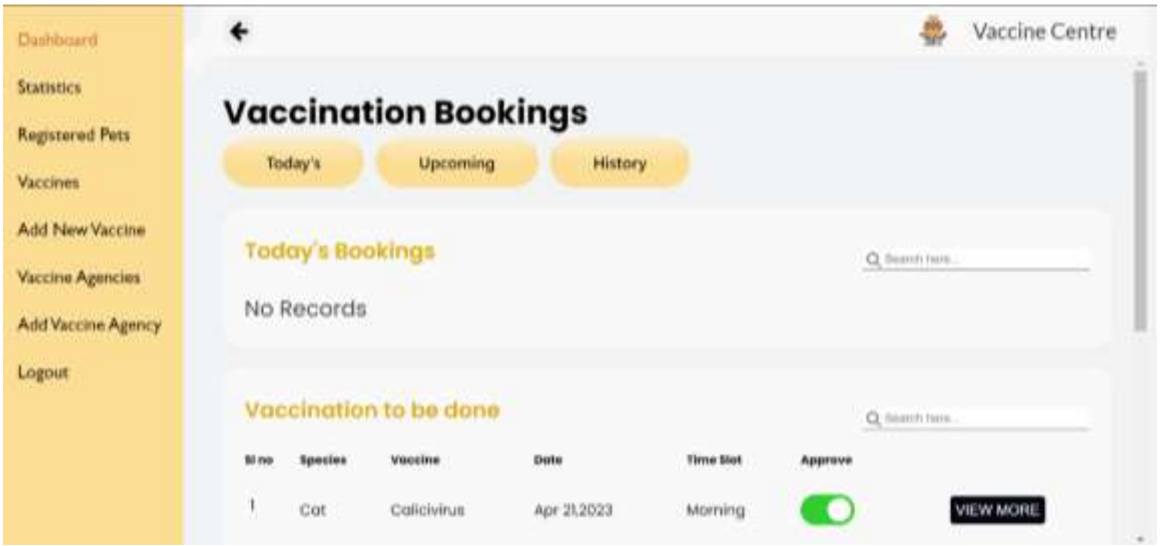
Sl no	Customer	Date		Status
1	Lakshmi Sunil	2023-03-31	VIEW MORE	ORDER CANCELLED
2	Celine Carmel M J	2023-03-16	VIEW MORE	PRODUCT DELIVERED

1.2.10 Admin – Sentiment Analysis Graph



Sentiment Analysis

1.2.11 Vaccine Center - Dashboard



Vaccination Bookings

Today's Bookings

No Records

Vaccination to be done

Sl no	Species	Vaccine	Date	Time Slot	Approve	
1	Cat	Calicivirus	Apr 21,2023	Morning	●	VIEW MORE