**WOOF & WAG**

PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF THE

REQUIREMENTS FOR THE AWARD OF

**BACHELOR OF COMPUTER APPLICATIONS**

To

**MARIAN COLLEGE KUTTIKKANAM[AUTONOMOUS]**

Affiliated to

**MAHATMA GANDHI UNIVERSITY, KOTTAYAM**

By

**DEVU SUGATHAN**

(Reg.No:22UBC223)

Guided By

**Ms. JEENA JOSEPH**



**DEPARTMENT OF COMPUTER APPLICATIONS**

**MARIAN COLLEGE KUTTIKKANAM (AUTONOMOUS)**

**PEERMADE – 685531**

**OCTOBER, 2024**

**DECLARATION**

I, **DEVU SUGATHAN [Reg.no:22UBC223]** certify that the Mini project report entitled **“WOOF & WAG”** is an authentic work carried out by me at Marian College Kuttikkanam (Autonomous). The matter embodied in this project work has not been submitted elsewhere for the award of any degree or diploma to the best of my knowledge and belief.

Signature of the Student:

Name of the Student: **DEVU SUGATHAN**

Date:

**BONAFIDE CERTIFICATE**

This is to certify that this project work entitled **“WOOF & WAG”** is a bonafied record of work done **by Ms. DEVU SUGATHAN [Reg.no:22UBC223]** at Marian College Kuttikkanam (Autonomous) in partial fulfilment for the award of the **Degree of Bachelor of Computer Applications of Mahatma Gandhi University, Kottayam.**

This work has not been submitted elsewhere for the award of any degree to the best of my knowledge.

Head of the Department Internal Guide

**Dr Rajimol A** **Ms. Jeena Joseph**

Dept. of Computer Applications Dept. Computer Applications

Marian College Kuttikkanam (Autonomous) Marian College Kutiikkanam (Autonomous)

Peermade – 685531 Peermade – 685531

Submitted for the Viva-Voce Examination held on

**DEPARTMENT SEAL**

**EXTERNAL EXAMINAR**

**ACKNOWLEDGEMENT**

“Gratitude is a feeling which is more eloquent than words, more silent than silence.” In undertaking this project, I needed the direction, assistance and cooperation of various individuals and organizations, which is received in abundance with grace of God, without their unconstrained support, the project could not have been completed.

If words are considered as the symbol of approval and token of acknowledgement, then let the following words play the heralding role of expressing my gratitude.

I wish to acknowledge my sincere gratitude to our Manager, **Rev.Fr. Boby Alex Mannamplackal and Dr Ajimon George, Principal Marian College Kuttikkanam (Autonomous),** for all their efforts and administration in educating me in their premier institution.

I extend my gratitude to **Dr Rajimol A,** Head of the Department of Computer Application, who is a constant source of inspiration and whose advice helped me to complete this project successfully.

I express my deep sense of gratitude to my internal project guide**, Ms. Jeena Joseph**, for her profound guidance for the successful completion of this project.

With great enthusiasm I express my gratitude to all the faculty members of Department of Computer Applications for their timely help and support.

Finally, I express my deep appreciation to all my friends and family members for the moral support and encouragement they have given to complete this project successfully.

Devu Sugathan

**ABSTRACT**

**ABSTRACT**

"Woof & Wag" is a pet care website designed to enhance the welfare of dogs with no shelter through health management and community involvement. The website encourages community engagement by allowing users to report sightings. Pet day care, pet grooming and donations for pets are also available in ‘woof & wag’.

The user can report if any dog is in danger through the form given on the website, and the user can log in to the website to get updates about his or her report. If a user wants to donate money to dogs, they can donate through the website. If a user needs pet day care or pet grooming services, they can book through the website and log in to check whether the booking is accepted or if any further notification is there. In the case of pet day care, the user will get updates about the dog. The user can provide a username and password after registration or report.

Admins can go through all the functionalities of the website. An Admin can give updates to the user about rescuing a dog after getting any reports from the user. After rescuing a dog, the admin can register the details of the new dog in the database of ‘woof &wag’. The admin is responsible for managing the registration of pet grooming, pet day care and donations, which includes checking the dates, making payments, and notifying the user about all the updates, etc.

This website helps to create a safer environment for both street dogs and communities. The site also features a user-friendly interface, allowing users to easily contact pet care. The website is developed using HTML, CSS, JavaScript, and PHP, which collectively contribute to the creation of a well-developed, dynamic website.

**Table of Contents**

**TITLE**

**1.ABSTRACT**

1.1 ABOUT THE PROJECT

1.2 EXISTING SYSTEM

1.3 PROPOSED SYSTEM

**2. SYSTEM ANALYSIS**

2.1 PROBLEM DEFINITION

2.2 FEASILIBILITY ANALYSIS

2.3 RECOMMENDED IMPLEMENTATIONS

**3. SOFTWARE REQUIREMENT SPECIFICATION**

3.1 INTRODUCTION

3.2 PURPOSE

3.3 SCOPES

3.4 TECHNICAL OVERVIEW

3.5 STATED REQUIREMENTS

3.6 EXTERNAL INTERFACE REQUIREMENTS

**4. SYSTEM DESIGN**

4.1 INTRODUCTION

4.2 DESIGN METHODOLOGY

PAGE NUMBER

6

10

10

10

10

13

13

13

14

17

17

17

17

18

18

20

23

23

4.3 SYSTEM ARCHIETECTURE AND PROCESS FLOW

4.4 MODULE DETAILS

4.5 PERFORMANCE ISSUES

4.6 SECURITY ISSUES

4.7 TABLE DESIGN

24

29

30

33

34

**INTRODUCTION**

**1. INTRODUCTION**

**1.1 ABOUT THE PROJECT**

Woof & Wag is a comprehensive platform dedicated to the welfare of street dogs, allowing users to report dogs in danger and donate to support the cause. The website also offers premium services such as pet grooming and pet daycare, enabling users to register and make payments seamlessly. Through Woof & Wag, you can ensure the safety and care of street dogs while enjoying top-notch pet services.

**1.1.1 THE PURPOSE AND SCOPE**

The Woof & Wag project aims to create a user-friendly platform that promotes the welfare of street dogs by allowing users to report dogs in danger, make donations, and access premium pet grooming and daycare services. The project seeks to foster a compassionate community dedicated to ensuring the safety and well-being of street dogs while providing top-quality care for pets.

* Develop a feature for users to easily report street dogs in danger, ensuring swift assistance and rescue.
* Implement a secure and efficient system for users to donate money, supporting the organization's rescue and care efforts.
* Offer an online booking and payment system for professional grooming services to keep pets clean and healthy.
* Provide a booking and payment system for pet daycare services, ensuring pets are well-cared for while their owners are away
* Create a user-friendly registration and profile management system to streamline access to all services and features on the website..

**1.2 EXISTING SYSTEM**

Woof & Wag stands out by offering a comprehensive platform that not only allows users to report street dogs in danger but also facilitates donations directly supporting dog welfare. Unlike existing systems, Woof & Wag integrates premium pet grooming and daycare services with easy registration and secure payment options. This holistic approach ensures both street dogs and pets receive the best care. Additionally, the user-friendly interface and seamless navigation make it more accessible and efficient for users to contribute and access services compared to other platforms.

**1.3 PROPOSED SYSTEM**

The primary objective of the Woof & Wag project is to establish a comprehensive platform dedicated to the welfare of street dogs and the provision of premium pet services. This includes creating an efficient system for users to report street dogs in danger and facilitating secure donations to support rescue efforts. Additionally, the project aims to offer top-quality pet grooming and daycare services, ensuring pets receive exceptional care. By simplifying user registration, booking, and payment processes, Woof & Wag enhances overall user experience and convenience. Ultimately, the project seeks to build a compassionate community focused on the safety and well-being of both street dogs and pets.

**SYSTEM ANALYSIS**

**2. SYSTEM ANALYSIS**

**2.1 PROBLEM DEFINITION**

The Woof & Wag project addresses the critical issue of street dog welfare by providing a platform where users can report dogs in danger, which often goes unnoticed in many communities. Existing systems lack a unified approach to rescue, care, and user engagement, making it difficult for concerned individuals to contribute effectively. Additionally, the need for premium pet grooming and daycare services is often unmet due to fragmented service offerings. Woof & Wag aims to solve these problems by integrating rescue reporting, donations, and pet care services into a single, user-friendly platform.

**2.1.1 ADVANTAGES OF PROPOSED SYSTEM**

* Users can easily report street dogs in danger, ensuring timely assistance and rescue efforts.
* A secure and efficient platform for donations supports the organization's mission, making it easier for users to contribute to dog welfare.
* Offers premium pet grooming and daycare services, providing a one-stop solution for all pet care needs.
* Simplifies registration, booking, and payment processes, enhancing convenience and user satisfaction.
* Builds a compassionate community dedicated to the safety and well-being of street dogs and pets, fostering greater social responsibility.

**2.2 FEASIBILITY ANALYSIS**

Feasibility study is a test of a system proposal according to its workability, ability to meet user needs and effective use of resources. The objective of feasibility is not to solve the problem but to acquire a sense of its scope. The main aim of the feasibility study is to test the technical, social and economic feasibility of the system. The feasibility study can be classified into the following categories:

• Operational Feasibility

• Technical Feasibility

• Economic Feasibility

**2.2.1 OPERATIONAL FEASIBILITY**

The proposed system offers user friendliness combined with greater processing speed. Therefore, the work done can be reduced. Since the processing speed is very high compared 14 with manual system a lot of time can be saved. The workload is reduced and this system requires only a small amount of work from Admin who manages the whole system.Hence, this project is operationally feasible.

**2.2.2 TECHNICAL FEASIBILITY**

Technical feasibility deals with hardware as well as software requirements and to what extend it can support the proposed system. The hardware required is an android phone and software is Android Studio. If the necessary requirements are made available with the system, then the proposed system is said to be technically feasible.

**2.2.3 ECONOMIC FEASIBILITY**

Economic feasibility is an important factor. Since the existing system is manual on the feasibility for wrong data entry is higher and consumes a lot of time and can occur errors. But the proposed system aims at processing of information’s efficiently, thus saving the time. The new system need only a system and which is already available therefore the cost is negligible. Proposed system uses validation check so there are no errors. Even though an initial investment has to be made on the software and the hardware aspects, the proposed system aims at processing of information’s efficiently. Thus, the benefits acquired out of the system are sufficient enough for the project to be undertaken. So, the proposed system is economically feasible.

**2.3 RECOMMENDED IMPLEMENTATIONS**

Two principle sources of data are:

1. Written documents

2. Data from the persons, who are involved in the operation of the system under study.

The different fact-finding techniques are:

1. Questionnaires

2. Personal Interviews

3. Observations

**Questionnaires**

Questionnaires are best methods to probe data out of the customers. In this case, questionnaires were not used for data collection as the administration was small in number and they could be asked questions in a more effective interview.

**Personal Interviews**

Personal interviews are the best way to gather facts. This was the primary source of fact finding used for this project. The owner and the employees were interviewed and data collected. They were asked how the administrative duties and record keeping happened under the existing system. And suggestions were taken on what they wanted to add in addition in this system. Almost all their suggestions were integrated into this project.

**Observations**

A person can understand a lot about a system just by observing it. By being a bystander and observing how a day passes in the real estate helped to kick off this project. Using this method resulted in a better understanding of the workings of the organisation and what to do to make this an web application.

Observing the current system, one can understand that a lot of paperwork and staff services involved in the administrative and distribution area. The way in which the records are kept gives an idea for a strong database model.

**SOFTWARE REQUIREMENT SPECIFICATION**

**3.SOFTWARE REQUIREMENT SPECIFICATION**

**3.1 INTRODUCTION**

Requirements specification is the starting step for the development activities. It is currently one of the weak areas of software engineering. During requirement specification, the goal is to produce a document of the client’s requirements. This document forms the basis of development and software validation. The basic reason for the difficulty in software requirements specification comes from the fact that there are three interested parties- the client, the end users and the software developer.

**3.2 PURPOSE**

The origin of most software systems is in the need of a client, who either wants to automate an existing manual system or desires a new software system. The software system itself is created by the developer. Finally, the completed system will be used by the end users. Thus, there are three major parties interested in a new system: the client, the users and the developer. A basic purpose of software requirements specification is to bridge the communication gap. SRS is the medium through which the client and user needs are accurately specified. Indeed, SRS forms the basis of software development. A good SRS should satisfy all the parties, something very hard to achieve, and involves trade-offs and persuasion.

Another important purpose of developing an SRS is helping the clients understand their own needs. Advantages are:

* An SRS establishes the basis for agreement between the client and the supplier on what the software product will do.
* An SRS provides a reference for validation of the final product.
* A high-quality SRS is a prerequisite to high-quality software.
* A high-quality SRS reduces the development cost.

**3.3 SCOPES**

**3.3.1 SYSTEM STATEMENT OF SCOPE**

Woof & Wag is a pet care organization dedicated to ensuring the welfare of street dogs through user-reported data and offering services like pet day care and grooming. Users can donate to support the cause and receive prompt responses from the admin regarding grooming, day care registrations, and dog welfare reports.

**3.4 TECHNICAL OVERVIEW**

The system can be accessed by two types of users, the Admin and User. The admin has access only to the admin dashboard. The Customer have access to the site and all its services. The admin is invisible to user.

**3.5 STATED REQUIREMENTS**

**3.5.1 GENERAL REQUIREMENTS**

The system has 7 functional modules divided by admin and users.

**1.Login**

* Only registered users, and the admin can login to system to avail the service.
* The registered users, and admin uses their username and password to login.
* The username and password should always be valid.
* Password can also contain both upper case and lower case alphabetic, numbers and special character.
* Admin will be redirected to the admin panel when login with the predefined username and password.
* User will be redirected to the home page when login with the predefined username and password.

**2.Sign Up**

The new users need to register in order to get services like pet grooming, pet day care, donating money, and reporting sightings and to avail all the services. This includes several fields:

-Name -Password -Confirm Password

**3.Admin Panel**

The admin panel contains the different admin processes like view new reports from user, new registrations for pet grooming, pet day care, and donating money, accepting or rejecting registrations, and adding new admin.

**4. Home page**

• The main page of the woof & wag site where customers are taken first.

• Only registered customers can view this home page and can sign up or login to get more services.

• When user needs services, if the user not logged in, the service will not be enabled. Only logged customers can register or report sightings.

**5.Reporting**

The user can report the sightings, if any street dog is in danger. Before reporting, the user should log in. Place details, user details, reason for reporting, and dog details are taken for reporting. User can check whether the report is accepted or not.

**6.Pet day care**

The user can register for day care service. Before reporting, the user should log in. Date, user details, and dog details are taken for registering. User can check whether the registration is accepted or not, and also check the updates of dog.

**7.Pet grooming**

The user can register for pet grooming service. Before reporting, the user should log in. Date, user details, and dog details are taken for registering. User can check whether the report is accepted or not.

**8.Donation**

The user can donate money for dogs. Before donating, the user should log in. User details, are taken for donating.

**9.payment**

The users can do their payment for pet grooming, pet day care or can donate money. Bank

details, user details are taken for this.

**3.5.2 INPUTS**

The Woof & Wag Website will take the information about street dogs and done the rescue. Pet grooming, pet day care, Donation informations for registration are taken and done payment.

**3.5.3 PROCESSING**

• All types of validation for the data entry are carried out.

•Checking new donations and reports, Accepting/Rejecting registration for pet grooming, and pet day care, and payments are managed by admin.

• registrations for pet day care, pet grooming, donation, and reporting sightings are done by users.

• registering and getting the service.

• Payment

**3.5.4 OUTPUTS**

This system produces the following outputs:

* Users receive confirmations and updates abouts their dog after giving their dogs for day care.
* Woof & Wag provides prompt assistance for street dogs in danger based on user reports.
* Users receive registration status after registering for pet day care, pet grooming, donation.

**3.6 EXTERNAL INTERFACE REQUIREMENTS**

**3.6.1 USER INTERFACES**

All user interfaces will be GUI interfaces. Interfaces are designed to use with ease and without any confusion. The user interface shall have a pleasing appearance and high functionality.

• Suitable design and pleasing colours are selected to design the window page to make the users comfortable to operate the software.

• Component like textboxes and buttons are selected to make it easy to fill with appropriate data.

**3.6.2 HARDWARE INTERFACES**

The System needs a computer or any other smart phones or devices with network availability to browse into the web application. No other external hardware is required.

**HARDWARE SPECIFICATION**

Processor : Intel Pentium or higher

RAM : 256 MB or higher

Hard disk drive : 100MB is required on disk

Keyboard : Standard QWERTY keyboard

**IMPLEMENTATION SPECIFICATION**

Operating system : Windows OS

**3.6.3 SOFTWARE INTERFACES**

**SOFTWARE SPECIFICATION**

Operating System : Windows 11

DBMS : Mysql

Tool Used : PHP

**SYSTEM DESIGN**

**4. SYSTEM DESIGN**

**4.1 INTRODUCTION**

The purpose of design phase is to plan a solution of the problem specified by the analysis phase. This phase is the first step in moving from the problem domain to solution domain.

System design describes the desired features and operation in detail, including screen layouts, business rules, process diagrams, pseudo code and other documentation.

In this phase, the software’s overall structure and its nuances are defined. In terms of the client/server technology, the number of tiers needed for the package architecture, input design, output design, the database design, the data structure design etc. are all defined in this phase. Analysis and design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. So much care is taken during this phase.

The logical system of the product and the physical characteristics of the system are designed during this phase. The operating environment is established, major resources. Everything requiring user input or approval must be documented and reviewed by the user. The physical characteristics of the system are specified and a detailed design is prepared.

The subsystem identified during design are used to create a detailed structure of the system. Each subsystem is partitioned into one or more design units or modules. Detailed logic specifications are prepared for each software module. The logic of the module is usually specified in a high-level design description language, which is independent of the target language in which the software will eventually be implemented.

A good design must consider:

• Prompt: should be simple and clear to intuitively lead the user to an expected outcome.

• Memory load: Studies show that, under normal circumstances, users have a short-term memory of approximately six words. Ideally, the number of choices of users to select should be four or less. Otherwise, callers become confused and forget the choices presented to them.

• Service reaches ability: It is not pleasant for a person to go through a large number of steps before he reaches a service. Users starts to get impatient with more than five steps. Minimize the number of steps a caller must take to reduce frustration.

• Navigation: Provide a way to navigate back and forth between various dialogue steps. The user should be able to go to different parts of the dialog easily.

• Phonetic similarity: Provide a clear set of choices for user to select. Avoid choices with similar pronunciation.

• Error handling: Humans make mistakes. Graceful error handling decreases dependency on operators.

• User update: Let the user know what is going on and keep him engaged.

For the general design one or more potential design are propose and broadly sketched. Then these alternatives are presented to the users, who choose the design that best suits their requirements while staying with in the project constraints.

Detailed design stage is specification for the user interface, database, programs, hardware, and training and system documentation. Several structured techniques are used during the design phase. To design the software components, the designer transforms the automated processes in the physical data flow diagram into a program structure chart, which decomposes software processes into detailed modules and shows control path between modules

**4.2 DESIGN METHODOLOGY**

**4.2.1 INPUT DESIGN**

In the input design, the user oriented inputs are converted into computer recognizable format. The collection of input data is the most expensive part of the system in terms of equipment used, time and number of users involved. Input design is the processes of converting user oriented inputs to a computer based format. The goal of designing input data is to make data entry as easy, logical and free from errors as possible.

Input design is the link between the information system and the users and the skip necessary to put transaction data in to a usable form for processing. Instructing the computer to read data from a written printed document can activate the activity of putting data into the computer for processing or it can occur by keying data directly into the system. The design of input focusing on controlling the amount of input required, controlling the errors, avoid delay extra steps, and keeping the process simple. System analysis decides the following input design details:

• What data to input

• What medium is to use

• How the data is arranged and coded.

• Data items and transaction needing validation to detect error occurs

Activities performed as part of input design are:

• Data recording

• Data verification

• Data conversion

• Data validation

• Data correction

**4.2.1 OUTPUT DESIGN**

Output design is a process that involves designing necessary outputs that have to be used by various users according to requirements. Designing computer should proceed in well thought out manner. The term output means any information produced by the information system whether printed or displayed. When analyst design computer output they identified the specific output that is needed to meet the requirement.

Computer is the most important source of information to the users. Efficient intelligent output design should improve the system relationship with the user and help in decision making. When designing the output, system analyst must accomplish the following:

• Determine the information to present

• Decide whether to display, print, speak the information and select the output medium

• Arrange the information acceptable format

The output design is the key to the success of any system. Output is the key between the user and the sensor. The output must be concerned to the system’s working, as it should. Output design consists of displaying specification and procedures as data presentation. User is never left with the confusion as to what is happening without appropriate error and acknowledges message being received

**4.2.3 CODE DESIGN**

The coding step is a process that transform design into programming language. It translates a detail design representation of software into a programming language realization. The code design should be done in such a way that the lines of code used in the software should be minimum for the specified design of the solution. The coding should be in modularized manner.

When code is placed in a module, one may hide it from view and give those executable statements a name (the name of the function or procedure). Information hiding is a good thing when it enhances the understanding of a program by letting to focus on a higher level of abstraction. Information hiding is a bad thing when it obscures one’s understanding of a program. This usually happens when the name for the module is not chosen accurately.

In this software, the modularized approach is used. Different functions are created for different operations. The name of the module is chosen such a way that it describes what it does, i.e. the name gives the action performed by the module

**4.2.4 DATABASE DESIGN**

The details about the relevant data that came into lay in the system are identified according to the relationship the tables are designed by following the standard database design methods. The dative for each data in the table is defined. For optimum design of database to have better response time, to have data integrity, to avoid the redundancy and for security of the database tables created and analysed.

A database system can be defined as a representation of an information system in a computer. The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the user. In database design, several specific objectives are considered:

• Controlled redundancy

• Ease of learning and use

• Data independence

• More information at low cos

• Accuracy and integrity

• Recovery from failure

• Privacy and security

• Performance

The scheme is the view that helps us the DBMS decide what data in storage it should act upon as requested by the web application program. The subschema is concerned with a relatively small part of scheme. In database design, several views of data must be considered along with the persons who use them. The logical view is what the data look like, regardless of how they stored. The physical view is the way data exists in physical storage. It deals with how data are stored, accessed or related to other data in storage. The logical view are the users view the programmer’s view and the overall logical view, called a schema.

A **primary key** is a special relational database table column (or combination of columns) designated to uniquely identify all table records. A **primary key's** main features are: It must contain a unique value for each row of data. It cannot contain null values. There is different primary key for varies tables. For eg: in login, l\_id is the primary key, in reporting phone is the primary key so on.

**Normalization** is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

This project has used a main database having different tables, based on which the operations can perform well.

**4.3 SYSTEM ARCHITECTURE AND PROCESS FLOW**

**UML Diagram**

**4.3.1 Use Cases**

|  |  |
| --- | --- |
| Use case Id: |  |
| Use case name: | Sign Up |
| Created by: |  |
| Date created: |  |
| Description: | Allows users to create a new account on Woof & Wag |
| Primary actor: | User |
| Secondary actor: | None |
| Precondition: | The user is on the login page and does not have an existing account. |
| Postcondition | The user successfully creates a new Woof & Wag account and get an id also. |
| Main flow: | 1. The user selects the "Sign Up" option.  2. The system prompts for necessary information (username, password).  3. The user provides the required information and submits the registration form.  4. The system creates a new user account and get an id as an alert.  5. The use case ends. |

• Use Case for Woof & Wag Sign Up

|  |  |
| --- | --- |
| Use case Id: |  |
| Use case name: | login |
| Created by: |  |
| Date created: |  |
| Description: | Allows users to log in to existing account on Woof & Wag. |
| Primary actor: | User |
| Secondary actor: | None |
| Precondition: | The user is on the login page and does have an existing account. |
| Postcondition | The user successfully log in to his Woof & Wag account and the user will enter into the home page. |
| Main flow: | 1. The system prompts for necessary information (username, password).  2. The user provides the required information and submits the login form.  3. The user enters into home page. |

• Use Case for Woof & Wag login

• Use Case for Woof & Wag Reporting

|  |  |
| --- | --- |
| Use case Id: |  |
| Use case name: | Report |
| Created by: |  |
| Date created: |  |
| Description: | Allows users to report their sightings of street dogs to Woof & Wag |
| Primary actor: | User |
| Secondary actor: | None |
| Precondition: | The user is on the home page and wants to report his sightings. |
| Postcondition | The user successfully reports his sightings to Woof & Wag. |
| Main flow: | 1. The system prompts for necessary information (username, id, phone, place, etc).  2. The user provides the required information and submits the report form. |

• Use Case for Woof & Wag Pet grooming

|  |  |
| --- | --- |
| Use case Id: |  |
| Use case name: | Pet grooming |
| Created by: |  |
| Date created: |  |
| Description: | It allows users to book for pet grooming at Woof & Wag and can check the status after booking. |
| Primary actor: | User |
| Secondary actor: | None |
| Precondition: | The user is on the home page and wants to book for pet grooming and wants to check the status of booking. |
| Postcondition | The user successfully books pet grooming at Woof & Wag or the user will get the current status of booking. |
| Main flow: | 1. The system prompts for necessary information for booking (username, phone, id, breed, date, etc) or for checking status(id).  2. The user provides the required information and submits the form.  3.If it is for checking the status, the user will get the current status. |

• Use Case for Woof & Wag Pet day care

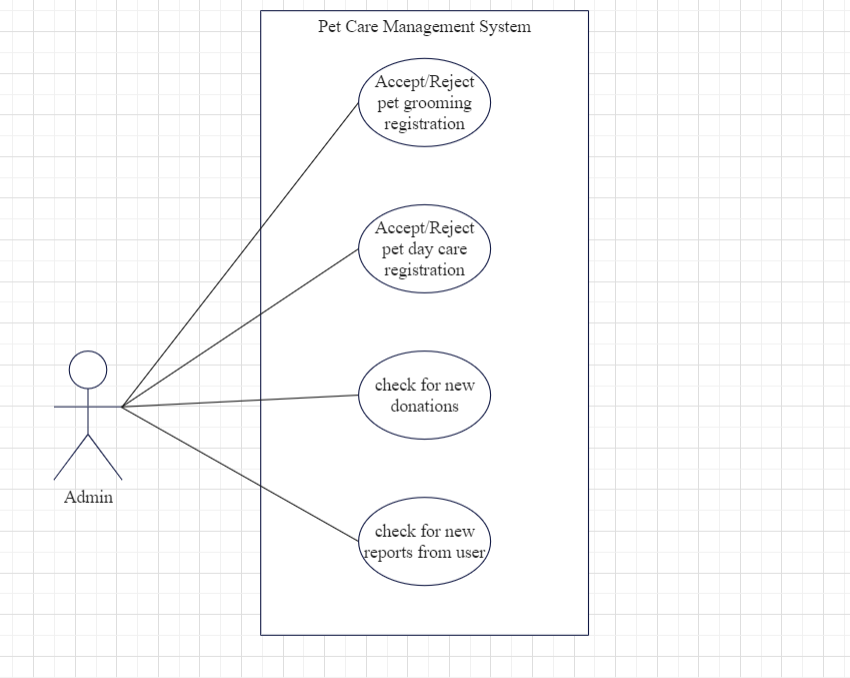
|  |  |
| --- | --- |
| Use case Id: |  |
| Use case name: | Pet day care |
| Created by: |  |
| Date created: |  |
| Description: | It allows users to book for pet day care at Woof & Wag and can check the status after booking. |
| Primary actor: | User |
| Secondary actor: | None |
| Precondition: | The user is on the home page and wants to book for pet day care and wants to check the status of booking. |
| Postcondition | The user successfully books pet day care at Woof & Wag or the user will get the current status of booking. |
| Main flow: | 1. The system prompts for necessary information (username, phone, id, breed, date, etc) or for checking status(id).  2. The user provides the required information and submits the form.  3.If it is for checking the status, the user will get the current status. |

• Use Case for Woof & Wag Donation

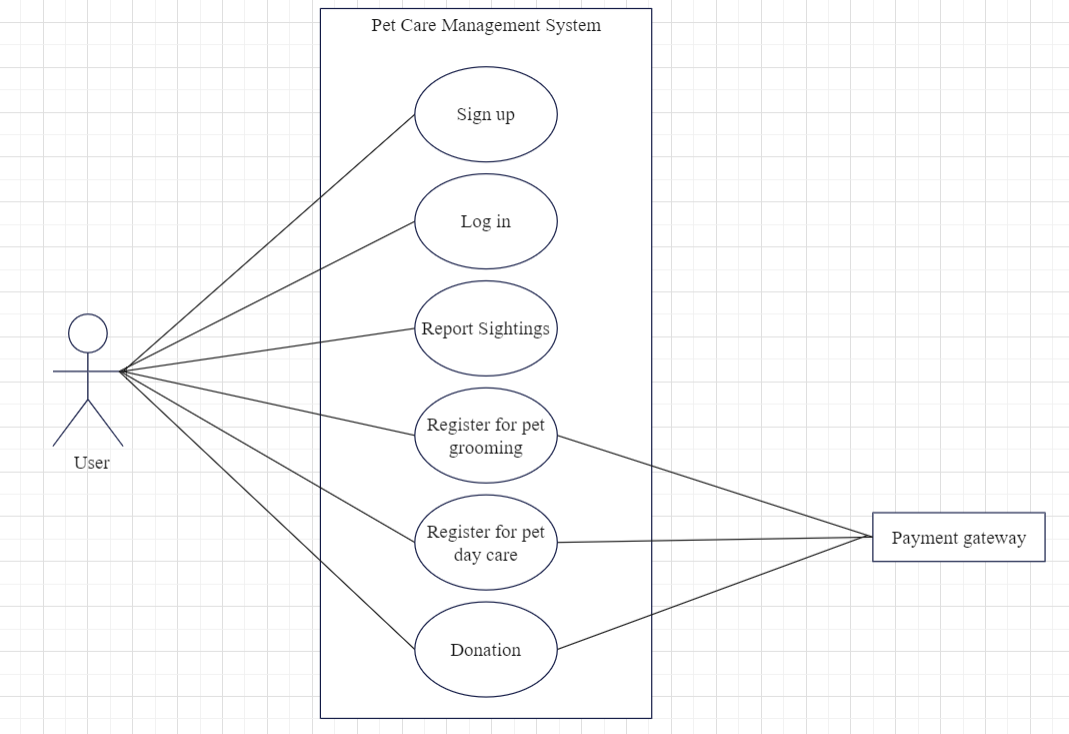
|  |  |
| --- | --- |
| Use case Id: |  |
| Use case name: | Donation |
| Created by: |  |
| Date created: |  |
| Description: | It allows users to donate money for street dogs at Woof & Wag |
| Primary actor: | User |
| Secondary actor: | None |
| Precondition: | The user is on the home page and wants to donate. |
| Postcondition | The user successfully donate money. |
| Main flow: | 1. The system prompts for necessary information (username, phone, id, etc)  2. The user provides the required information and submits the form. |

**4.3.2 Use case diagram**

**Admin side**

****

**User side**

****

**4.4 MODULE DETAILS**

This software has 9 module and different parts. They are:

* Login
* Sign Up
* Payment
* Admin Dashboard
* Home page
* Report
* Pet day care
* Pet grooming
* Donation

The login module provides provision for the registered customers to login to the site and can avail the services.

The Sign-up module is used to create new account for the customers for availing all the services provided.

The Admin tasks and several processes undertaken by the admin are made in this module.

The different modules and their functions are explained in detail in the above system requirements.

**4.5 PERFORMANCE ISSUES**

This system should have at least 4GB of RAM. The OS must be Windows or higher.

**4.6 SECURITY ISSUES**

Only authorised access is permitted to this software. The login is protected by username and password.

**4.7 TABLE DESIGN**

1. Table Name: login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | id | int | Auto\_increment,  Primary Key | Admin / User id |
| 2 | f\_name | varchar(50) | Not null | First name of the admin/user |
| 3 | l\_name | varchar(50) | Not null | Last name of the admin/user |
| 4 | email | varchar(150) | Not null | Email id of the admin/user |
| 5 | password | varchar(50) | Not null,  unique | Password of the admin/user |
| 6 | role | int | Default | Role of the person |

2. Table Name: paid\_services

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | s\_id | int | Auto\_increment,  Primary Key | Service id |
| 2 | s\_name | varchar(100) | Not null | Name of the service |
| 3 | amount | int | Not null | Per day amount of service |

3. Table Name: breed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | br\_id | int | Auto\_increment,  Primary Key | breed id |
| 2 | br\_name | varchar(100) | Not null | Name of the breed |

4. Table Name: reports

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | r\_id | int | Primary Key,Auto increment | report id |
| 2 | u\_id | int | Foreign Key  Not null | id of the user |
| 3 | r\_date | date | Not null | Reporting date |
| 4 | place | varchar(150) | Not null | Place where the dog is present |
| 5 | landmark | varchar(150) | Not null | Landmark of the place |
| 6 | reason | varchar(500) | Not null | Reason for reporting |
| 7 | color | varchar(50) | Not null | Color of the dog |
| 8 | r\_status | varchar(150) | Not null,  Default | Report status |
| 9 | r\_discription | varchar(300) | Not null,  Default | Description of report status |
| 10 | dog\_status | varchar(500) | Not null,  Default | Description of dog current status |

5. Table Name: donation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | don\_id | int | Auto\_increment,  Primary Key | donation id |
| 2 | u\_id | int | Foreign Key  Not null | id of the user |
| 3 | amount | int | Not null | Donating amount |
| 4 | don\_date | date | Not null | Donating date |

6. Table Name: grooming

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | g\_id | int | Auto\_increment,  Primary Key | grooming id |
| 2 | u\_id | int | Foreign Key  Not null | Id of the user |
| 3 | breed\_id | int | Foreign Key,  Not null | id of the breed |
| 4 | g\_date | date | Not null | Requested grooming date |
| 5 | g\_status | varchar(50) | Not null,  Default | Request status |
| 6 | g\_discription | varchar(50) | Not null,  Default | Request status discription |

7. Table Name: daycare

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | d\_id | int | Auto\_increment,  Primary Key | Day care id |
| 2 | u\_id | int | Foreign Key  Not null | id of the user |
| 3 | breed\_id | int | Foreign Key,  Not null | id of the breed |
| 4 | d\_starting\_date | date | Not null | Requested date to start day care service |
| 5 | d\_ending\_date | date | Not null | Requested date to end day care service |
| 6 | d\_status | varchar(150) | Not null,  Default | Request status |
| 7 | d\_discription | varchar(250) | Not null,  Default | Request status discription |
| 8 | pet\_status | varchar(500) | Not null,  Default | Updates of the pet |

8. Table Name: payments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | payment\_id | int | Auto\_increment,  Primary Key | Payment id |
| 2 | u\_id | int | Foreign Key | User id |
| 3 | service\_id | int | Foreign Key | Service id |
| 4 | user\_service\_id | int | Not null | ID of the user for a particular service |
| 5 | payment\_date | date | Not null | Payment date |

9. Table Name: bank

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | b\_id | int | Auto\_increment,  Primary Key | Bank id |
| 2 | acc\_num | int | Not null | Account number of the user |
| 3 | ifsc | varchar(150) | Not null | IFSC code of the bank |
| 4 | branch | varchar(100) | Not null | Branch name of the bank |
| 5 | balance | int | Not null | Account balance of the user |
| 6 | c\_num | int | Not null | Card number of the user |
| 7 | Pin | int | Not null | Pin code of the card |
| 8 | cvv | int | Not null | cvv number of the user |
| 9 | exp\_date | date | Not null | Expiry date of the card |