



DogLoversBD.com: A WEB APPLICATION FOR ONLINE DOG BUY & SELL

Prepared By:

MD. ASHIKUR RAHMAN KHAN

ID: 152-35-1185

BSc in Software Engineering

Daffodil International University

Supervised By:

KHALID BEEN MD. BADRUZZAMAN BIPLOP

Lecturer

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

DECLARATION

I hereby declare that; this project has been done by me under the supervision of **KHALID BEEN MD. BADRUZZAMAN BIPLOB, Lecturer, Department of Software Engineering, Daffodil International University**. I also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

ASHIK

Md. Ashikur Rahman Khan

ID: 152-35-1185

Department of Software Engineering

Daffodil International University

Certified by:

Badrizzaman 5.5.19

Khalid Been Md. Badruzzaman Biolop

Lecturer

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Acknowledgment

First of all, I would like to thanks and also very much grateful to my supervisor **Khalid Been Md. Badruzzaman Boplop, Lecturer, Department of Software Engineering, Daffodil International University**. Deep Knowledge & keen interest of our supervisor in the field of “A web app for online dog buy and sell” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express our heartiest gratitude to **Pro. Dr. Touhid Bhuiyan, Professor, and Head, Department of SWE**, for his kind help to finish my project and also to other faculty member and the staff of SWE department of Daffodil International University.

I would like to thank my entire course mate at Daffodil International University, who took part in this discussion while completing the course work. Finally, I must acknowledge with due respect the constant support and patience of my parents.

EXECUTIVE SUMMARY

My project titled “**DogLoversBD.com: A WEB APPLICATION FOR ONLINE DOG BUY AND SELL**” is a web-based app for online dog e-commerce of Bangladesh. Using this application, people can buy their choice able dog. Also, they can sell their dog through my website.

At present, there are many dog lovers’ people around our country. But we don’t have any fully web-based app for dog buy and sell it online. For this reason, the dog lovers’ people are confused about how to sell their dog or how to buy a dog in a proper way.

Nowadays the internet becomes a part of our life. We can’t think a single moment without it. Among these OS, Web has a majority percentage. For this reason, I thought that I should build something in Web that has a practical use to the user.

My project is based on the purpose of buy and sells the dog. Because there is a big community of dog lovers stands in Bangladesh. They need a platform by which they can maintain or also taken care of their dog.

My app will help with all the information about dog health tips and breed category. This will increase the dog lover’s community and decrease harassment. It also reduces time. This app is very much user-friendly. After finishing all the task and test process this application proved to be working effectively.

I have used Bootstrap, HTML, CSS, JavaScript for front-end design and validations and PHP programming language with Laravel Framework 5.8.10 for back-end design. I also used phpMyAdmin local Server as my main database.

TABLE OF CONTENTS

Declaration -----	i
Acknowledgment-----	ii
Executive Summary -----	iii
Table of Contents -----	iv
List of Figures -----	vii
List of Tables -----	1
CHAPTER 1: INTRODUCTION	
1.1 Project Overview -----	2
1.2 Project Purpose -----	2
1.2.1 Background -----	2
1.2.2 Benefits -----	3
1.2.3 Goals -----	3
1.3 Stakeholders -----	3
1.4 Proposed System Model (block diagram) -----	4
1.5 Project Schedule -----	5
1.5.1 Gantt Chart -----	5
1.5.2 Milestone -----	6
CHAPTER 2: SOFTWARE REQUIREMENT SPECIFICATION	
2.1 Functional Requirements -----	7
2.2 Data Requirements -----	8
2.3 Performance Requirements -----	9
2.3.1 Speed and Latency Requirements -----	9
2.3.2 Accuracy Requirements -----	9
2.3.3 Capacity Requirements -----	9
2.4 Dependability Requirements -----	10
2.4.1 Reliability Requirements -----	10
2.4.2 Availability Requirements -----	10
2.4.3 Robustness or Fault-Tolerance Requirements -----	10
2.5 Maintainability and Supportability Requirements -----	11
2.5.1 Maintenance Requirements -----	11
2.5.2 Supportability Requirements -----	11
2.6 Security Requirements -----	11
2.6.1 Access Requirements -----	11
2.6.2 Integrity Requirements -----	11
2.6.3 Privacy Requirements -----	11
2.7 Usability and human-interaction requirements -----	11
2.7.1 Ease of Use Requirements -----	11
2.7.2 Personalization and Internationalization Requirements ---	12

2.7.3	Understandability and politeness Requirements -----	12
2.7.4	Accessibility Requirements -----	12
2.7.5	User Documentation Requirements -----	12
2.7.6	Training Requirements -----	12
2.8	Look and feel requirements -----	12
2.8.1	Appearance Requirements -----	12
2.8.2	Style Requirements -----	12
CHAPTER 3: SYSTEM ANALYSIS		
3.1	Use Case Diagram -----	14
3.2	Use Case Description -----	15
3.2.1	Owner message admin -----	15
3.2.2	The owner sends dog's info -----	16
3.2.3	Owner check website -----	17
3.2.4	Owner deal with customer -----	18
3.2.5	Owner sell dog -----	19
3.2.6	Customer visit website-----	19
3.2.7	Customer choose dog-----	20
3.2.8	Customer contact with owner-----	21
3.2.9	Customer negotiate the price with owner-----	22
3.2.10	Customer buy dog-----	23
3.2.11	Admin registration-----	24
3.2.12	Admin login-----	25
3.2.13	Admin add, delete, update dog's info-----	26
3.2.14	Admin response owner email-----	27
3.2.15	Admin add message option-----	28
3.2.16	Admin add Facebook link-----	29
3.2.17	Admin add to dog's breed category -----	30
3.2.18	Admin add to dog's health tips-----	31
3.3	Activity Diagram -----	32
3.3.1	Activity Diagram for admin -----	32
3.3.2	Activity Diagram for customer -----	33
3.3.3	Activity Diagram for owner -----	34
CHAPTER 4: SYSTEM DESIGN SPECIFICATION		
4.1	Class Responsibilities Collaboration (CRC) Cards -----	35
4.2	Sequence Diagram -----	36
4.2.1	Sequence Diagram for Admin -----	36
4.2.2	Sequence Diagram for customer -----	37
4.2.3	Sequence Diagram for owner -----	38
4.3	Database Design Diagram -----	39
4.4	Development Tools & Technology -----	40
4.4.1	User Interface Technology -----	40
4.4.2	Implementation Tools & Platforms -- -----	40

CHAPTER 5: SYSTEM TESTING

5.1	Testing Features -----	41
5.1.1	Features to be tested -----	41
5.1.2	Features not to be tested -----	41
5.2	Testing Strategies -----	41
5.2.1	Test Approach -----	41
5.2.2	Pass/Fail Criteria -----	42
5.2.3	Suspension and Resumption -----	42
5.3	Testing Environment (hardware/software requirements) -----	42
5.4	Test Case Descriptions -----	42
5.4.1	Test case for admin login-----	42
5.4.2	Test case for verify owner-----	43
5.4.3	Test case for customer message-----	43
5.4.4	Test case for customer buy dog-----	44
5.4.5	Test case for owner sell dog-----	44

CHAPTER 6: USER MANUAL

6.1	Home Page -----	45
6.2	Dog's info -----	46
6.3	Health page -----	47
6.4	Facebook contact page -----	49
6.5	Contact page -----	49
6.6	Admin login -----	50
6.7	Local host details -----	52
6.8	Dog breed details -----	52

CHAPTER 7: PROJECT SUMMARY

7.1	GitHub Link -----	53
7.2	Critical Evolution -----	53
7.3	Limitations -----	53
7.4	Obstacles & Achievements -----	53
7.5	Future Scope -----	53
7.6	Discussion and Conclusion -----	54

REFERENCES -----	55
------------------	----

LIST OF FIGURES

Figure 1.1:	DogLoversBD Block Diagram -----	4
Figure 1.2:	Gantt chart -----	5
Figure 3.1:	Use Case Diagram -----	14
Figure 3.1:	Activity Diagram for admin -----	32
Figure 3.2:	Activity Diagram for customer -----	33
Figure 3.3:	Activity Diagram for owner -----	34
Figure 4.1:	Sequence Diagram for admin -----	36
Figure 4.2:	Sequence Diagram for customer-----	37
Figure 4.3:	Sequence Diagram for owner -----	38
Figure 4.4:	database design diagram-----	39
Figure 6.1:	Home page -----	45
Figure 6.2:	home page -----	45
Figure 6.3:	home page -----	46
Figure 6.4:	dog's info page -----	46
Figure 6.5:	dog's info page -----	47
Figure 6.6:	dog's health page -----	47
Figure 6.7:	dog's health page -----	48
Figure 6.8:	dog's health page -----	48
Figure 6.9:	dog's health page -----	49
Figure 6.10:	contact page -----	49
Figure 6.11:	contact page -----	50
Figure 6.12:	admin login -----	50
Figure 6.13:	admin contact list -----	51
Figure 6.14:	message list details -----	51
Figure 6.15:	phpMyAdmin local host -----	52
Figure 6.16:	breed list category -----	52

LIST OF TABLES

Table 1.1:	Milestone -----	6
Table 2.1:	Admin Registration -----	7
Table 2.2:	Owner mail -----	7
Table 2.3:	Buy dog -----	8
Table 2.4:	Health tips -----	8
Table 2.5:	Negotiate price -----	8
Table 2.6:	Speed and Latency Requirements -----	9
Table 2.7:	Accuracy Requirements -----	9
Table 2.8:	Capacity Requirements -----	9
Table 2.9:	Reliability Requirements -----	10
Table 2.10:	Availability Requirements -----	10
Table 2.11:	Robustness or Fault Tolerance requirements -----	10
Table 3.1:	Use Case Description of owner message admin -----	15
Table 3.2:	Use Case Description of owner send dog's info-----	16
Table 3.3:	Use Case Use Case Description of owner check website -----	17
Table 3.4:	Use Case Description of owner deal with customer -----	18
Table 3.5:	Use Case Description of owner sell dog -----	19
Table 3.6:	Use Case Description of customer visit website -----	19
Table 3.7:	Use Case Description of customer choose dog -----	20
Table 3.8:	Use Case Description of customer contact with owner -----	21
Table 3.9:	Use Case Description of customer negotiate the price -----	22
Table 3.10:	Use Case Description of customer buy dog -----	23
Table 3.11:	Use Case Description of admin registration -----	24
Table 3.12:	Use Case Description of admin login -----	25
Table 3.13:	Use Case Description of admin add, delete, update dog's info---	26
Table 3.14:	Use Case Description of admin response owner email -----	27
Table 3.15:	Use Case Description of admin add message option -----	28
Table 3.16:	Use Case Description of admin add Facebook link -----	29
Table 3.17:	Use Case Description of admin add to dog's breed category ----	30
Table 3.18:	Use Case Description of admin add to dog's health tips -----	31
Table 4.1:	Admin Responsibilities and collaboration -----	35
Table 4.2:	owner Responsibilities and collaboration -----	35
Table 4.3:	customer Responsibilities and collaboration -----	35
Table 5.1:	Test Case for Admin Login -----	42
Table 5.2:	Test Case for verifying owner -----	43
Table 5.3:	Test Case for customer message -----	43
Table 5.4:	Test Case for customer buy dog -----	44
Table 5.5:	Test Case for owner sell dog -----	44

CHAPTER 1

INTRODUCTION

1.1 Project Overview

A Web app can provide the solution by offering the ability to share issues and information between owner and customer which lead to rapid issue resolution and less misinterpretation. The goal of the web app is to create a better relationship between dog owner and customer.

The dogloversbd.com web application is designed to support both customer and owner. This project will help the customer to find the perfect dog. Also, this is a huge opportunity for the dog owner to sell their dog in a nice amount of price. For all the dog lovers this app will help them a lot.

1.2 Project Purpose

The main purpose of your project is to make a digital dog e-commerce system. Where customers can complete the buying process easier. Also, the dog owner can easily sell their dog. My project is for all those who have a dog. And who try to buy a choice dog.

1.2.1 Background

It is some web application a little bit similar to my web application but these have many differences with my web application. Some apps are only for dog accessories. But my project contains all the health tips and also breed information. But on the other hand, our dogloversbd.com give a customer or an owner all the necessary information about the dog's breed, health tips, buy, sell option. Also, we provide our Facebook group link and message option. That both customer and owner deal with each other so easily. Owner contact with admin to upload the dog's info which he/she send through email. Our admin seriously response all the owner request.

1.2.2 Benefits

Without any benefits, there is no fulfillment of any system. Our system has some unique benefits. They are given below:

- ▢ This is a user-friendly digitalized system.
- ▢ Customer can easily buy a dog through the system.
- ▢ The dog owner can easily sell the dog and receive a well amount of price.
- ▢ No one will be able to do any kind of harassment.

1.2.3 Goals

We want to make this web application. This web application will have contained these features.

- ▢ To keep the website user-friendly.
- ▢ To keep a legal way of business.
- ▢ To provide security to both customer and a dog owner.
- ▢ To maintain security in the business.
- Admin response owner email and update website, so that customer check dog's info.

1.3 Stakeholders:

A system must have some stakeholders. A system has no value without the relation of stakeholders with the system. Our system's stakeholders are:

Customer: Customer is the main internal stakeholder of our system. Because they will buy a dog.

Owner: dog owner is another main internal stakeholder of our system. The owner has the dog to sell.

Admin: Admin will be the main stakeholder. Admin has the authority or power to control the whole system. Owner email will be approved by the Admin. He will monitor the system and activity.

1.4 Proposed System Model (Block Diagram):

This is a proposed system of DogloversBD Block Diagram. [1]
When a digital complex system is showing through a diagram in a schematic way from the general arrangement then it is called a Block Diagram.

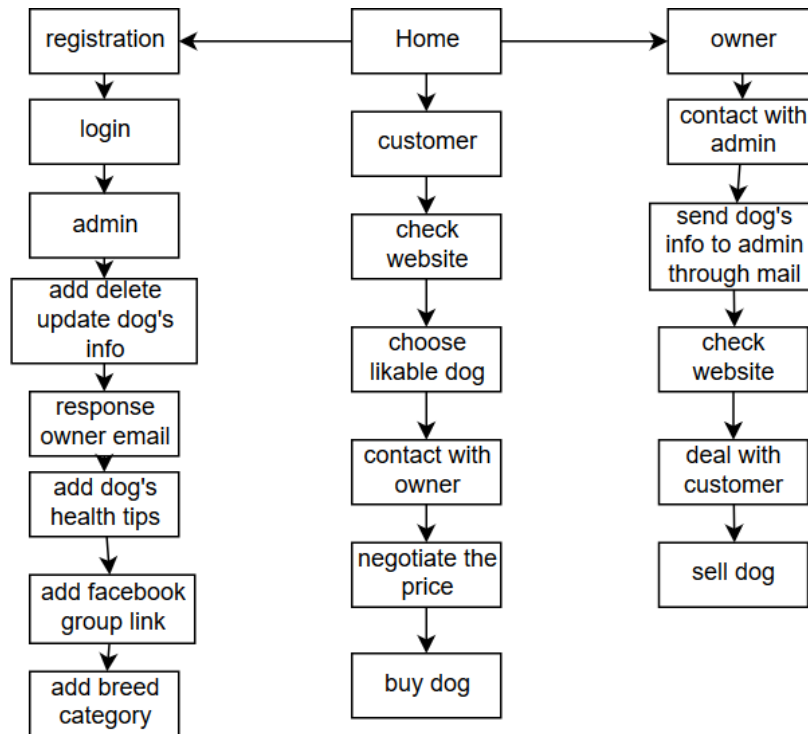


Figure 1.1: DogLoversBD Block Diagram

1.5 Project Schedule

Full filling the required requirements and complete the project in the time we maintain the project schedule. We make a project schedule to complete the project properly in time.

1.5.1 Gantt Chart

A Gantt chart[2] is a type of bar chart that illustrates a project schedule.

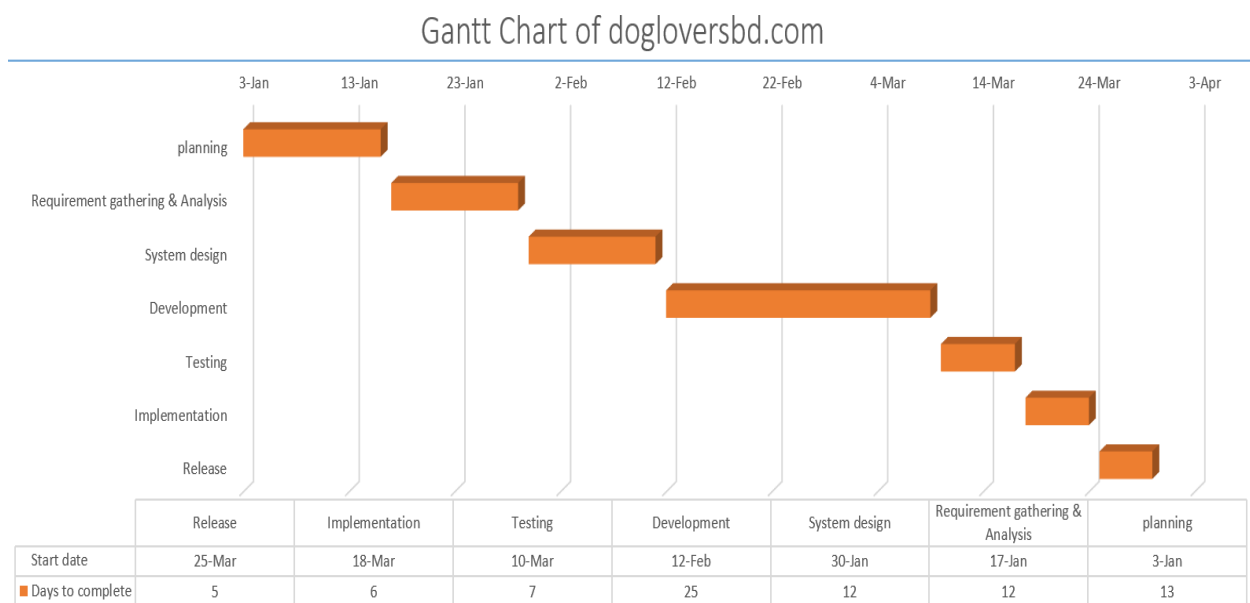


Figure 1.2: Gantt chart DogLoversBD.

1.5.2 Milestone:

This is a DogLoversBD project milestone that contains the project timeline. Milestone, it is a time frame of the project. That will define the project task. Project milestones are as follows:

Table 1.1: Milestone

Task no.	Task Name	Start Date	End Date	Duration
01	Planning	03-01-19	16-01-19	13 days
02	Requirements Gathering & Analysis	17-01-19	29-01-19	12 days
03	System Design	30-01-19	11-02-19	12 days
04	Development	12-02-19	09-03-19	25 days
05	Testing	10-03-19	17-03-19	07 days
06	Implementation	18-03-19	24-03-19	06 days
07	Release	25-03-19	30-03-19	05 days
	Total			80 days

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

Requirement analysis is the process of identifying user satisfaction in the most important part of project management.

2.1 Functional Requirements:

The functional requirement of the system is given below: this table contains the full system of DogLoversBD.

Admin Registration:

Table 2.1: Admin Registration

Fr-01	Admin Registration
Description	Admin can assign an owner and can monitor the whole system.
Stakeholder	Admin

Owner email:

Table 2.2: owner mail

Fr-02	Owner mail
Description	The owner can message into the system and can perform his define task.
Stakeholder	Owner

Buy dog:

Table 2.3: buy a dog

Fr -04	Buy dog
Description	Completing the investigation process with validity Customer buy dog from the system.
Stakeholders	Customer & owner.

Health tips:

Table 2.4: health tips

Fr -05	Health tips
Description	Customer finds health tips on the website and takes care of their dog. As well as an owner takes proper care to their dog. Admin adds health tips.
Stakeholders	Admin, customer, owner.

Negotiate deal:

Table 2.5: negotiate deal

Fr-06	Negotiate deal
Description	Customer contact with the owner and negotiate the deal.
Stakeholders	Owner & customer.

2.2 Data requirements:

In this stage, first, we have to understand the type of gathering data to enhance our system.

- ▯ Connecting with the central database.
- ▯ Information about owner mail verified.
- The owner sends the dog's info to admin.
- ▯ Admin upload data.
- ▯ Customer checks the info and contact with the owner.
- ▯ Deal information.

2.3 Performance Requirements:

It's very necessary to sustain the performance of the project. To assure the better performance, this project has to meet some requirements which will provide better performance.

2.3.1 Speed and Latency Requirements:

While the user will run our project in the browser, the system needs a minimum amount of speed to perform the task.

Table 2.6: Speed and Latency Requirements

SLR-01	The processing speed of the system will be faster.
Description	When user will run our system speed will be depended on their internet speed & the server bandwidth speed.
Stakeholders	Users as a customer, owner, Admin, etc.

2.3.2 Accuracy Requirements:

The system has to confirm the Legibility and Accuracy of the data.

Table 2.7: Accuracy Requirements

AR-01	Required proper information of every user
Description	The input data should be the correct and right pattern data, otherwise the input field show error message, like personal information, phone number, email address, password, etc. the input information is not valid, the data never save. Or the input data pattern is not match, then users will not be able to go forward step with the system.
Stakeholders	Users as a customer, owner, Admin, etc.

2.3.3 Capacity Requirements:

The system should maintain all inserting data

Table 2.8: Capacity Requirements

CR-01	Manages all the data in the database properly
Description	All data like owner personal data, dog's data, customer message data, etc. are stored in the database.
Stakeholders	Users as a customer, owner, Admin, etc.

2.4 Dependability Requirements:

Dependability means it measures system reliability, availability, robustness, safety, etc. Here, dependability means the running time of this project.

2.4.1 Reliability Requirements:

Table 2.9: Reliability Requirements

RR-01	The system is reliable
Description	Owner, customer information, other information will be safe and secured in our system.
Stakeholder	Users as owner, customer, Admin, etc.

2.4.2 Availability Requirements:

Table 2.10: Availability Requirements

AR-01	The system will be available 24x7.
Description	It is available 24 hours in a day and 7 days in a week The system will be updated regularly.
Stakeholder	Users as owner, customer, Admin, etc.

2.4.3 Robustness or Fault Tolerance requirements:

Table 2.11: Availability Requirements

FTR-01	Well robustness of the system
Description	If any problem occurs the system will show error message and the fault tolerance are handled properly.
Stakeholder	Users as owner, customer, Admin, etc.

2.5 Maintainability and supportability requirements:

Authority would have assigned some Admins and specialist for the maintenance of the system and support the system.

2.5.1 Maintenance Requirements:

Authority would have to assign some specialists for maintaining the system. The system can produce wrong results and the specialists must be able to reproduce the data flow through the system.

2.5.2 Supportability Requirements:

Authority would have to recruit some admins for supportability of the system.

2.6 Security requirements:

There are two security features as log in as Admin & access website as a customer or owner to get access to this system or a specific module of the system must provide an authentication mechanism.

2.6.1 Access Requirements:

This system provides accesses the different module, by access in an authentication way to the authentic user.

2.6.2 Integrity Requirements:

Credential information of users' will be kept safe.

2.6.3 Privacy Requirements:

All the passwords and personal information will be kept safe and securely in the system.

2.7 Usability and human-interaction requirements:

This system is very user-friendly with simple user-interface.

2.7.1 Ease of Use Requirements:

Users can use the system very easily with satisfaction and efficiency.

2.7.2 Personalization and Internationalization Requirements:

We developed this system for only Bangladeshi dog lovers' people. So, there are no internationalization requirements.

2.7.3 Understandability and politeness Requirements:

This system is very easy to use and understand with simple user-interface.

2.7.4 Accessibility Requirements:

This system is very easily accessible for the definite user in the definite module.

2.7.5 User Documentation Requirements:

Owner message information and records will be stored and kept in the system properly.

2.8 Look and feel requirements

According to the client's expectation look means the graphical user interface, design, layout, shapes, colors, etc. and feel means the behavior of different elements like the menu bar, buttons, etc.

2.8.1 Appearance Requirements

Needed knowledge about PHP with Laravel Framework and algorithm.

2.8.2 Style Requirements

Knowledge about CSS, JavaScript, and bootstrap.

Chapter 03: diagrams

CHAPTER 3

SYSTEM ANALYSIS

3.1 Use Case Diagram:

Use case Diagram is the simplest way of representing in a graphical representation of a complex system's user interactions among the elements of the system. Marriage Registrar's use case diagram is given below:

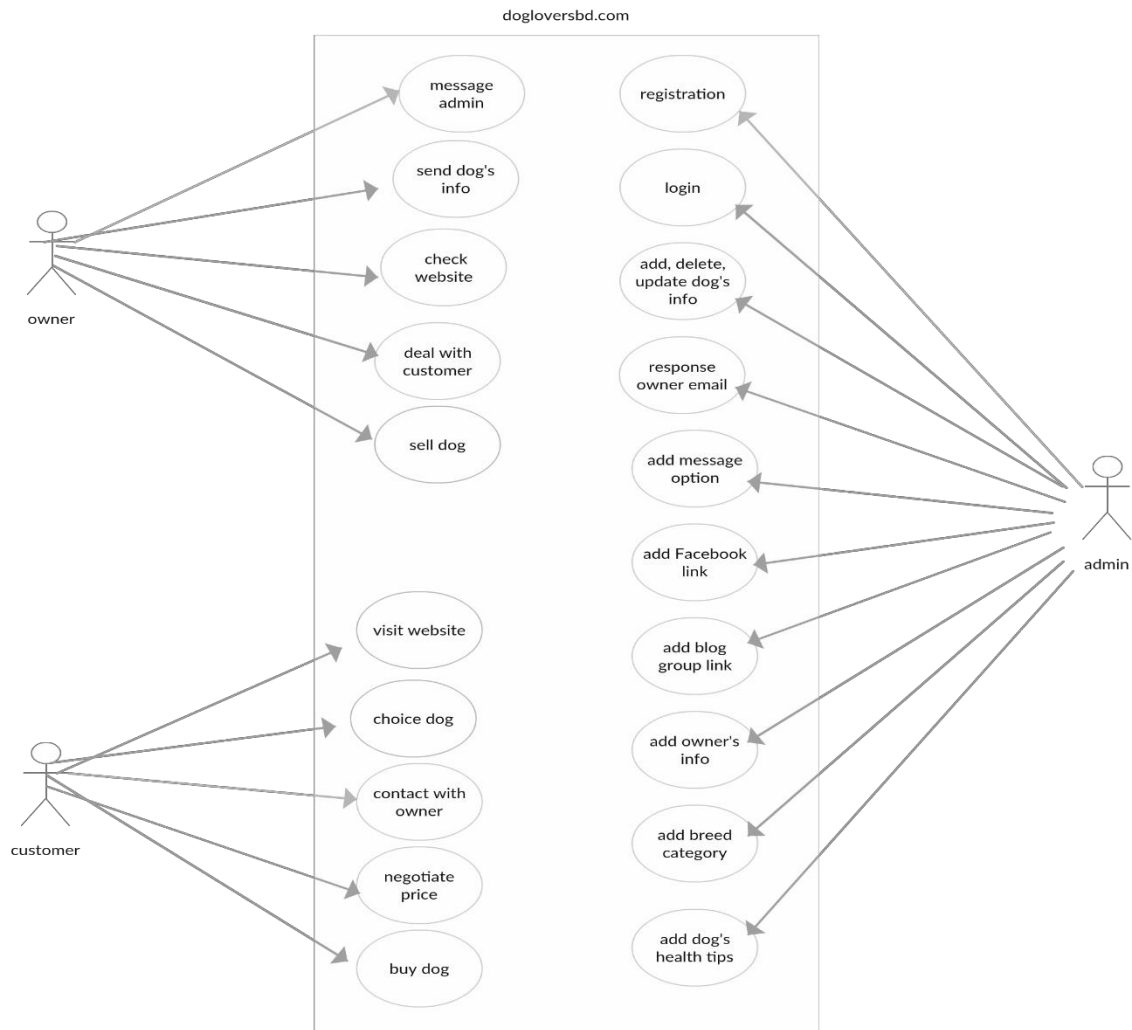


Figure 3.1 Use Case Diagram

3.2 Use Case Description:

In this part, there is described briefly how the users will execute tasks on the dogloversbd.com system.

3.2.1 Owner message admin:

Table 3.1: Use Case Description of owner message admin

Use case name	Owner message admin		
Scenario	Owner send message to admin to the platform		
Brief description	Only admin can see the message send by the dog's owner. This message owner asks an admin to response owner mail.		
Actor	Owner		
Preconditions	An owner must exist.		
Postconditions	Admin response to owner. Admin sends the owner a response message.		
Flow of events	Actor	System	
	The owner sends a message.	Response for a message.	
Failure results	If system not response then show nothing.		

3.2.2 Owner send dog's info:

Table 3.2: Use Case Description of owner send dog's info

Use case name	The owner sends the dog's info		
Scenario	Dog's owner sends admin his/her dog's info		
Brief description	When admin response owner email, then owner send the dog's info to the admin for uploading it in the website.		
Actor	Owner		
Preconditions	The owner must exist.		
Postconditions	The owner sends the dog's info to admin.		
Flow of events	Actor	System	
	Owner send Dog's info to admin.	Response with displaying proper information.	
Failure results	If system not response then show nothing.		

3.2.3 Owner check website:

Table 3.3: Use Case Description of owner check website

Use case name	Owner check website		
Scenario	After sending dog's info owner check website for surety.		
Brief description	Receiving the response message owner check website that whatever admin upload the dog's info.		
Actor	Owner		
Preconditions	The owner must exist.		
Postconditions	A successful message is shown by the system. Dog's info upload in website and everybody sees that.		
Flow of events	Actor	System	
	Owner check website for Investigating.	Response with displaying a successful message and	
Failure results	If system not response then show nothing.		

3.2.4 Owner deal with the customer:

Table 3.4: Use Case Description of owner deal with customer

Use case name	Owner deal with customer		
Scenario	Customer buys a dog so that the owner has to deal with Customer.		
Brief description	Customer check the website find the dog and deal with the An owner for price.		
Actor	Customer		
Preconditions	Customer exists.		
Postconditions	If customer want the dog so that he/she may deal with Dog's owner.		
Flow of events	Actor	System	
	Customer contact With owner for Buy dog.	Response with displaying proper information.	
Failure results	If system not response then show nothing.		

3.2.5 Owner sell dog:

Table 3.5: Use Case Description of owner sell dog

Use case name	Owner sell dog		
Scenario	Owner sells a dog to a customer.		
Brief	Owner negotiates with the price and sells a dog to a customer.		
Actor	Owner		
Preconditions	Customer must exist.		
Postconditions	If the owner deal with the customer and when it has been Successful, the owner sells the dog.		
Flow of events	Actor	System	
	owner deal with customer for Price and sell dog.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.6 Customer visit website:

Table 3.5: Use Case Description of customer visit website

Use case name	Customer visit website		
Scenario	Customer visit website for the searching dog.		
Brief	Customer visit website and find the perfect dog.		
Actor	Customer.		
Preconditions	the website must exist.		
Postconditions	Customer buy dog so that he/she check visit website If any choice able dog available.		
Flow of events	Actor	System	
	Customer Check Website For find dog.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.7 Customer choose dog:

Table 3.5: Use Case Description of customer choose a dog

Use case name	Customer choose dog		
Scenario	Customer visit website for choose dog.		
Brief	Customer visit website and find the perfect dog.		
Actor	Customer.		
Preconditions	the website must exist.		
Postconditions	Customer buy dog so that he/she check visit website If any choice able dog available.		
Flow of events	Actor	System	
	Customer Choose dog.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.8 Customer contact with the owner:

Table 3.5: Use Case Description of customer contact with the owner

Use case name	Customer contact with the owner		
Scenario	Customer visit website for choose dog and contact		
Brief	Customer visit website and find the perfect dog.		
Actor	Customer.		
Preconditions	The owner must exist.		
Postconditions	Customer buy dog so that he/she check visit website If any choice able dog available.		
Flow of events	Actor	System	
	Customer Choose dog.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.9 Customer negotiate the price with the owner:

Table 3.5: Use Case Description of customer negotiate the price with the owner

Use case name	Customer negotiate the price with the owner		
Scenario	Customer visit website for choose dog and contact with Owner.		
Brief	The customer chooses a dog and negotiates for the price.		
Actor	Customer.		
Preconditions	The owner must exist.		
Postconditions	Customer buys a dog so that he/she negotiated with the owner.		
Flow of events	Actor	System	
	Customer negotiate For price With dog Owner.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.10 customer buy dog:

Table 3.5: Use Case Description of customer buy dog

Use case name	Customer buy dog		
Scenario	Customer visit website for choose dog and contact with Owner and buy dog		
Brief	Customer choose a dog and negotiate for price and buy		
Actor	Customer.		
Preconditions	The owner must exist.		
Postconditions	Customer buy dog from owner.		
Flow of events	Actor	System	
	Customer negotiate For price With dog Owner.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.11 Admin registration:

Table 3.5: Use Case Description of admin registration

Use case name	Admin registration		
Scenario	Admin at first registration		
Brief	Admin first registration to the system		
Actor	Admin		
Preconditions	The website must exist.		
Postconditions	Admin register to the system		
Flow of events	Actor	System	
	Admin Register To the System.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.12 Admin login:

Table 3.5: Use Case Description of admin login

Use case name	Admin login		
Scenario	Admin login to the system		
Brief	Admin first registration then login to the system		
Actor	Admin		
Preconditions	The website must exist.		
Postconditions	Admin gives username and password to login system.		
Flow of events	Actor	System	
	Admin Register To the System then Log in to the system.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.13 Admin add delete update dog's info:

Table 3.5: Use Case Description of admin add, delete, update

Dog info to the system.

Use case name	Admin add, delete, update info to the system		
Scenario	Admin login to the system and add, delete, update dog's Info.		
Brief	Admin first registration then login to the system and then Receive dog's info from the owner so that admin can add, delete, update to the system.		
Actor	Admin		
Preconditions	The website must exist.		
Postconditions	Admin adds, delete, update dog's info.		
Flow of events	Actor	System	
	Admin Register To the System then Log in to the system then adds delete update dog's info.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.14 Admin response owner email:

Table 3.5: Use Case Description of admin response owner email

Dog info to the system.

Use case name	Admin response owner email		
Scenario	Admin response owner email		
Brief	Admin first registration then login to the system and then Response owner email		
Actor	Admin		
Preconditions	The owner must exist.		
Postconditions	Admin response existing owner email		
Flow of events	Actor	System	
	Admin Register To the System then Log in to the system then add delete update dog's info and response dog owner email	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.15 Admin add message option:

Table 3.5: Use Case Description of admin add message option

Dog info to the system.

Use case name	Admin add message option		
Scenario	Admin add message option to the system		
Brief	Admin add message option to the system so that customer and owner reach admin when they needed.		
Actor	Admin		
Preconditions	Owner and customer must exist.		
Postconditions	Admin adds a message option to the system.		
Flow of events	Actor	System	
	Admin add Message option to the system	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.16 Admin add Facebook link:

Table 3.5: Use Case Description of admin add a Facebook link

Dog info to the system.

Use case name	Admin add a Facebook link		
Scenario	Admin add a Facebook link to the system		
Brief	Admin add a Facebook link to the system so that customer and owner also connected with 24/7.		
Actor	Admin		
Preconditions	Owner and customer must exist.		
Postconditions	Admin adds a Facebook link to the system.		
Flow of events	Actor	System	
	Admin add Facebook link to the system	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.17 Admin add to dog's breed category:

Table 3.5: Use Case Description of admin add to the dog's breed category

Dog info to the system.

Use case name	Admin add to dog’s breed category		
Scenario	Admin add to the dog’s breed category to the system		
Brief	Admin add to the dog’s breed category to the system so that customer and owner get the information about the dog’s breeding.		
Actor	Admin		
Preconditions	Owner and customer must exist.		
Postconditions	Admin adds breed category to the system.		
Flow of events	Actor	System	
	Admin add breed category to the System.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.2.17 Admin add to dog's health tips:

Table 3.5: Use Case Description of admin add to dog's health tips

Dog info to the system.

Use case name	Admin add to dog’s health tips		
Scenario	Admin add to dog’s health tips to the system		
Brief	Admin add to dog’s health tips to the system so that customer and owner take proper care of their dog.		
Actor	Admin		
Preconditions	The system must exist.		
Postconditions	Admin adds to dog’s health tips to the system.		
Flow of events	Actor	System	
	Admin add Dog’s Health tips to System.	Response with displaying success message.	
Failure results	If system not response then show nothing.		

3.3 Activity Diagram:

Activity diagram is a very important diagram in UML. Basically, it is a flow chart. It can describe the flow of the activity in a graphical representation [3].

3.3.1 Activity Diagram for Admin:

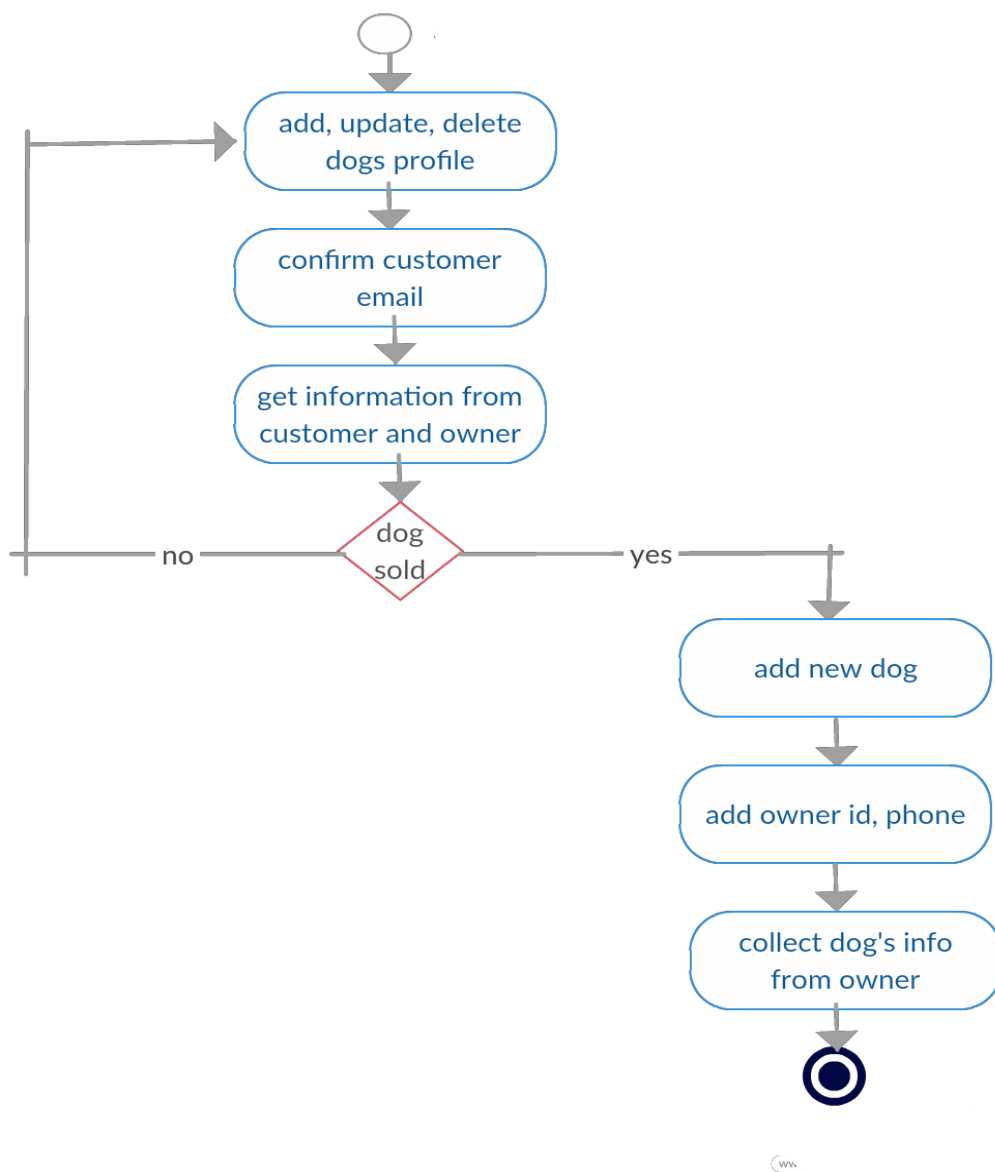


Figure 3.1: Activity Diagram for admin

3.3.2 Activity Diagram for the customer:

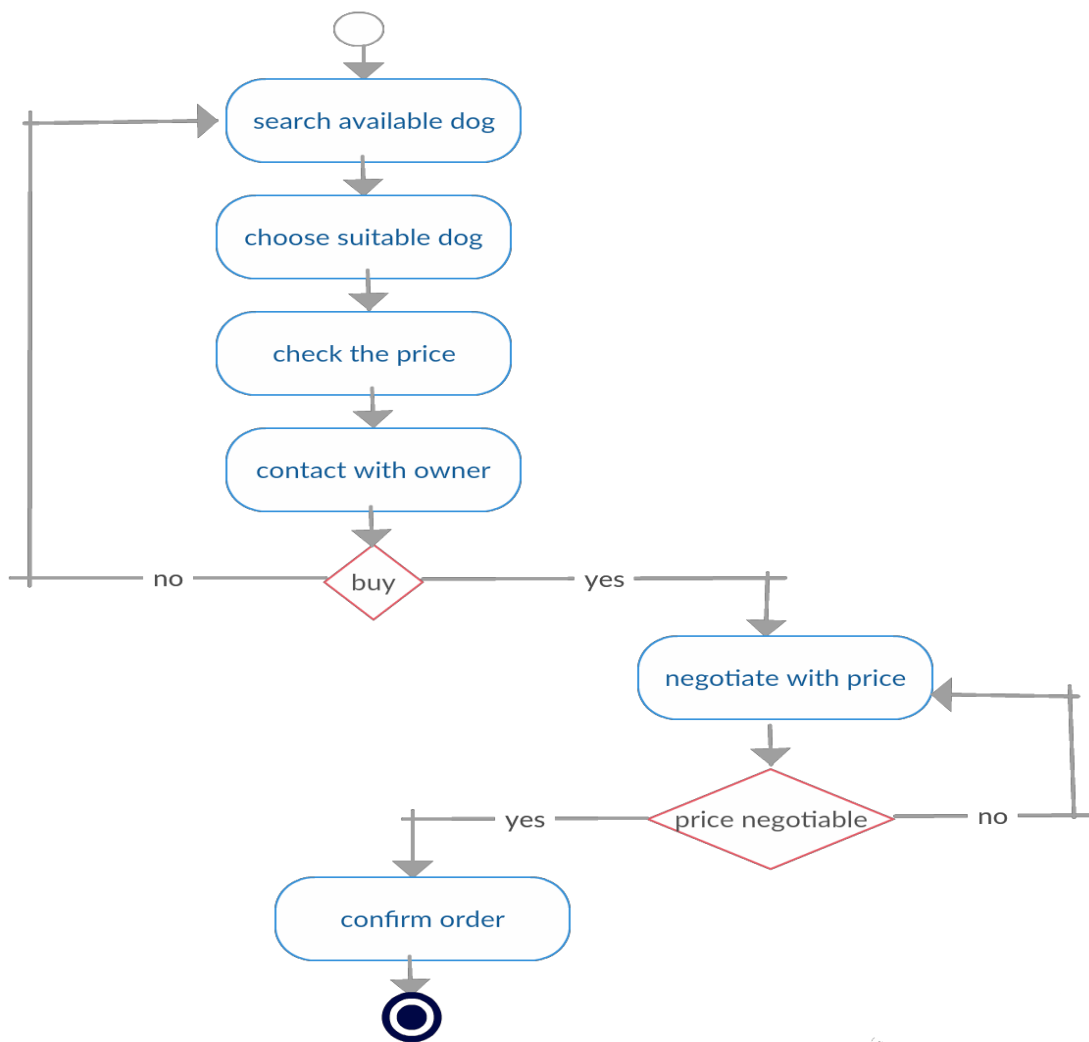


Figure 3.2: Activity Diagram for customer

3.3.3 Activity Diagram for owner:

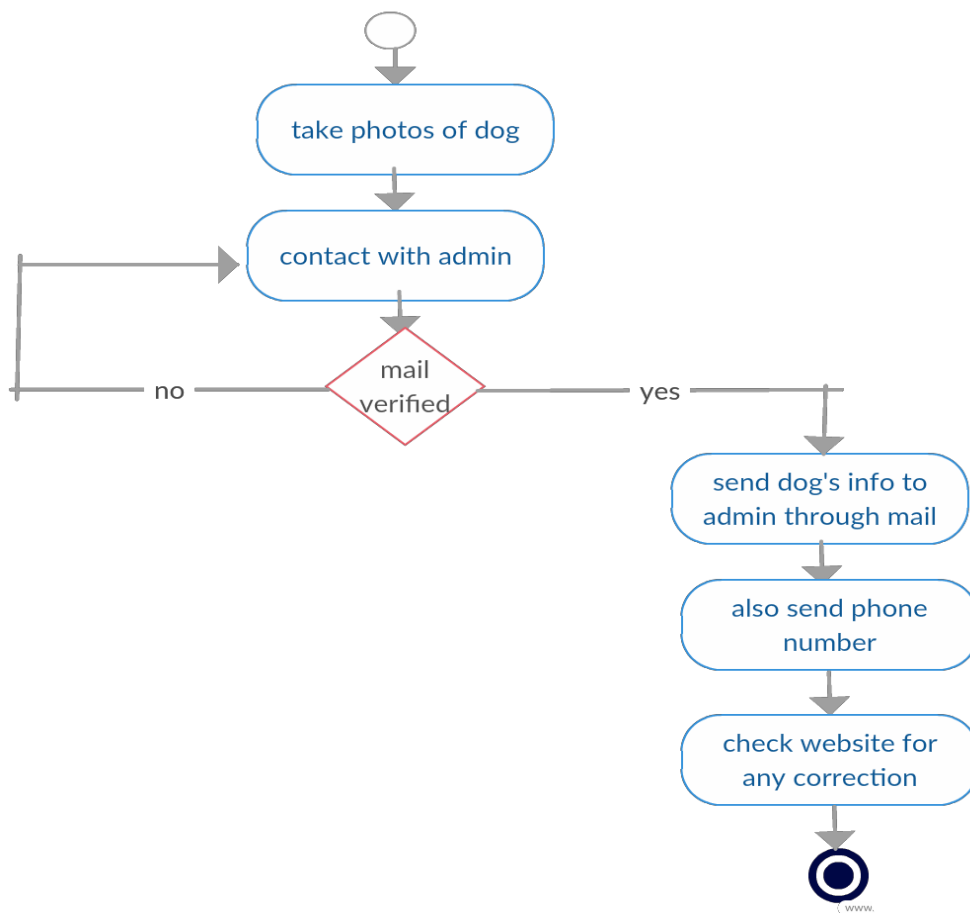


Figure 3.3: Activity Diagram for owner

CHAPTER 4

SYSTEM DESIGN SPECIFICATION

4.1 Class Responsibilities Collaboration (CRC) Cards:

It's a brainstorming tool used to design any kind of object-oriented software. Following tables are given below:

Admin:

Table 4.1: Admin Responsibilities and collaboration

Responsibilities	Collaboration
• Monitoring the system • Inquiry about regular tasks	• add, delete, update dog's bio

Owner:

Table 4.2: owner Responsibilities and collaboration

Responsibilities	Collaboration
• check website	• Send dog's info • Deal with customer

Customer:

Table 4.3: customer Responsibilities and collaboration

Responsibilities	Collaboration
• knowing the process of buying.	• negotiate the price with the owner.

4.2 Sequence Diagram:

A sequence diagram is another very important UML diagram of software. Sequence diagram can describe the working process of a system as to how it works and what order is following it to complete a process [4].

4.2.1 sequence diagram for admin:

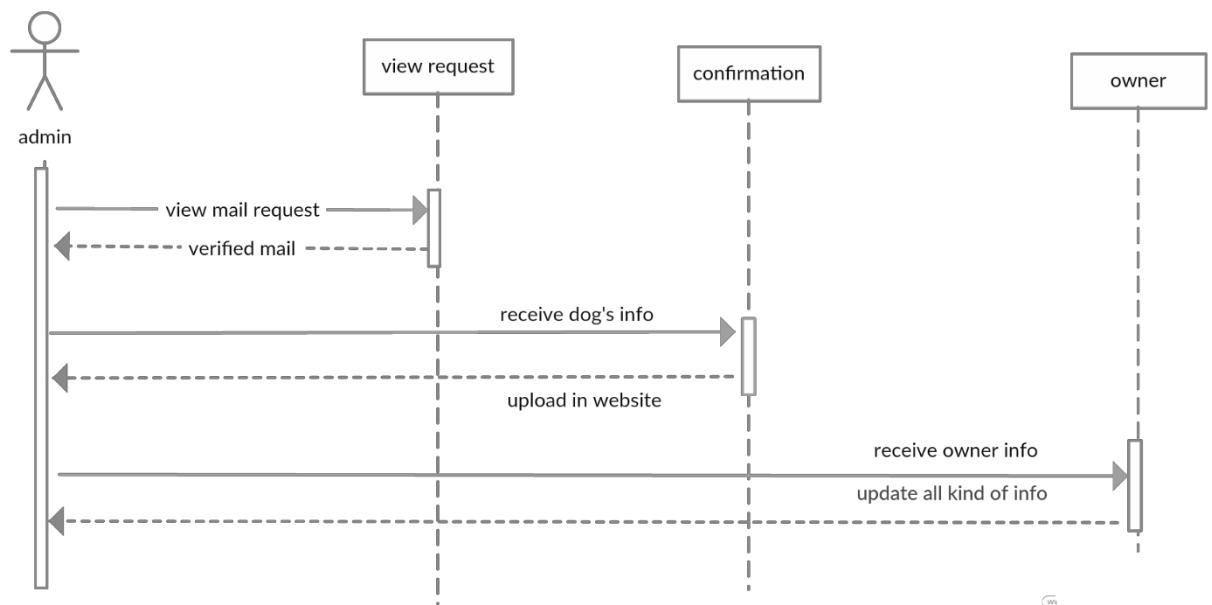


Figure 4.1: Sequence Diagram for admin

4.2.2 sequence diagram for customer:

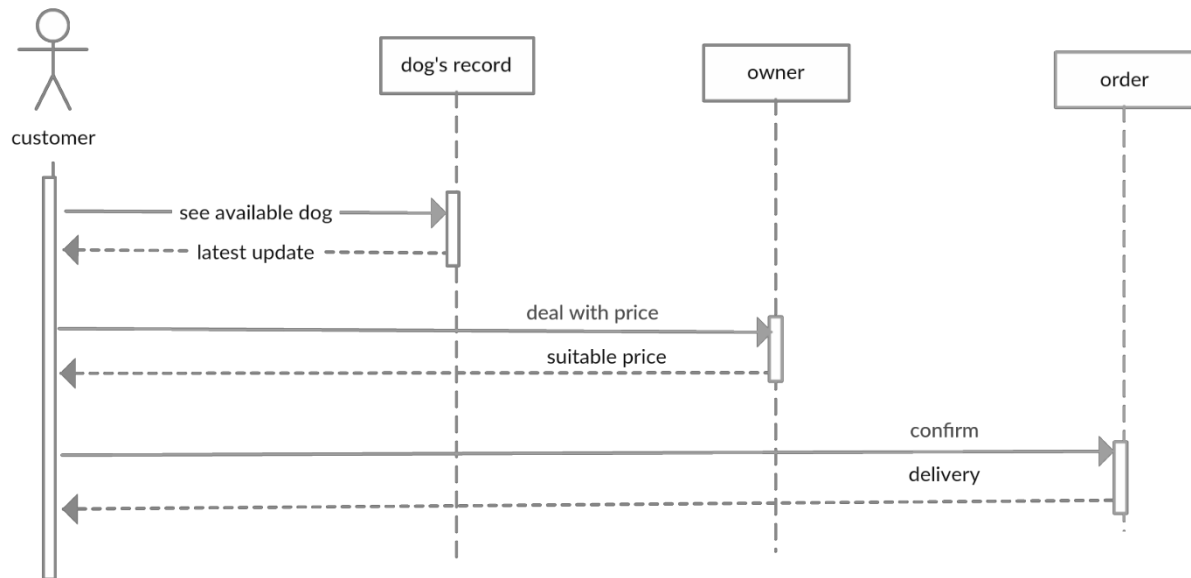


Figure 4.2: sequence Diagram for customer.

4.2.3 sequence diagram for owner:

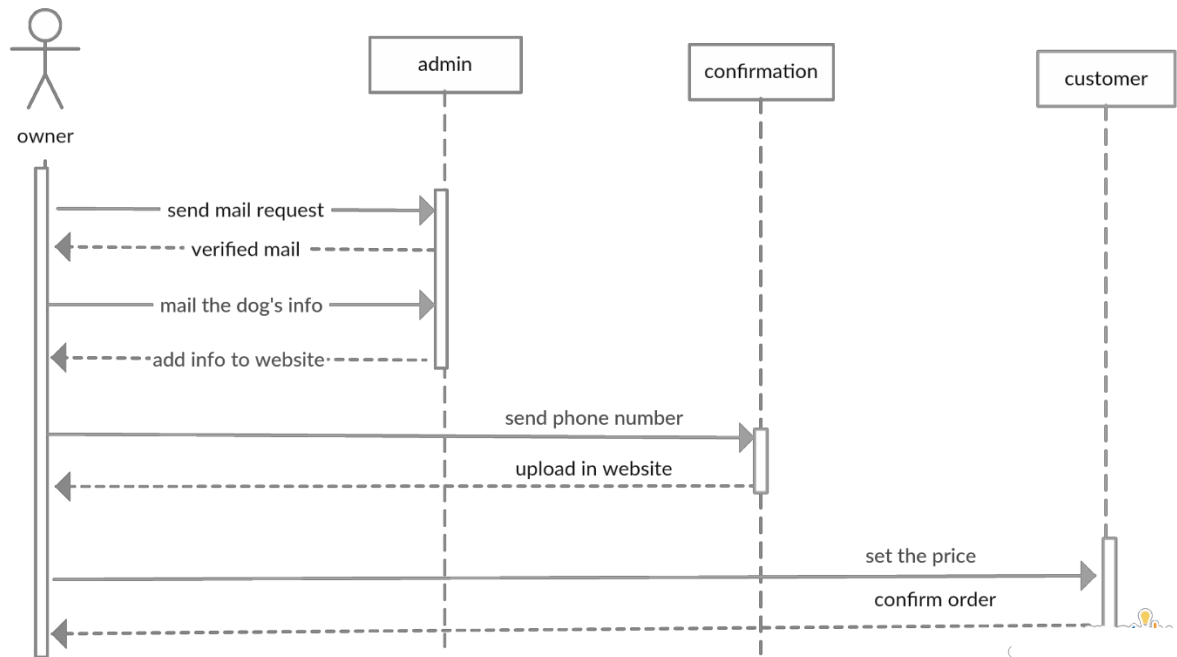


Figure 4.3: Sequence Diagram for the owner.

4.3 Database Design Diagram:

The database is the most important part of any system. We have tried to design a dynamic database for our system [5]. Our database design diagram is given below:

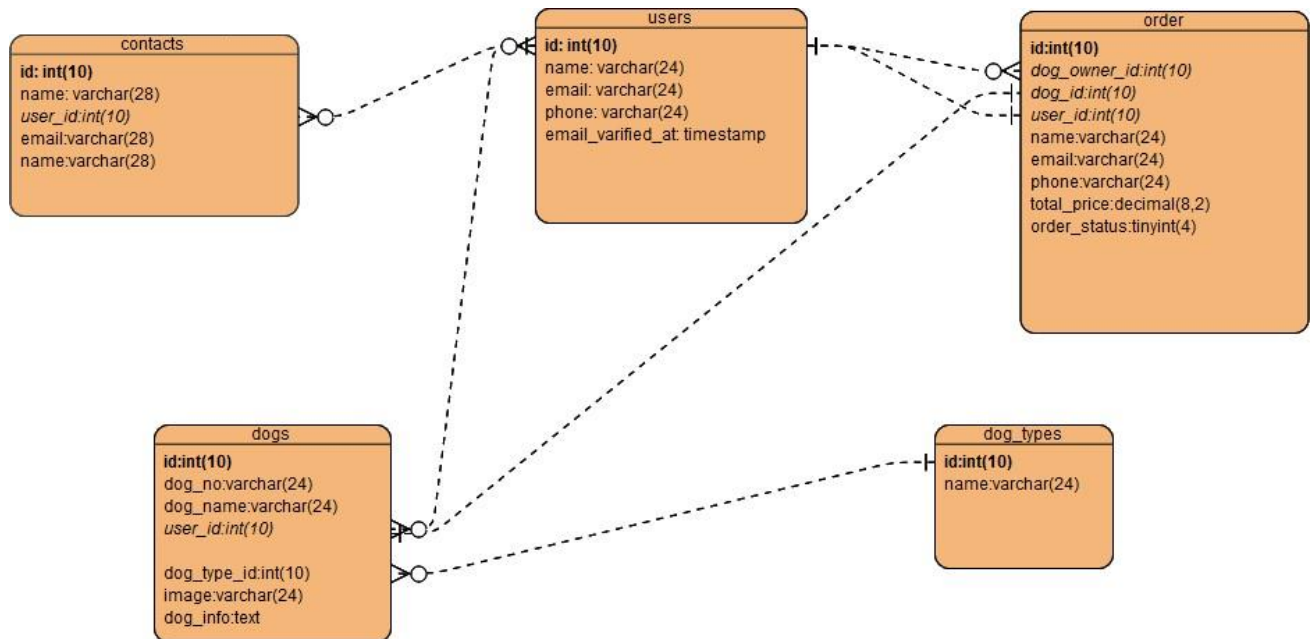


Figure 4.1: Database Design

4.2 Development Tools & Technology:

For developing a software system, selecting development tools and technology is a vital fact and challenge [6]

Development tools:

- ▯ Development IDE: PhpStorm
- ▯ Server: localhost server
- ▯ Operating System: Windows 10

Development technology:

- ▯ Programming language: PHP
- ▯ Pattern: Web API 2
- ▯ Framework: Laravel framework 5.8.10

4.2.1 User Interface Technology:

- ▯ HTML
- ▯ CSS
- ▯ Bootstrap
- ▯ JavaScript

4.3.2 Implementation tools & Platforms:

Implementation tools:

- ▯ Development IDE: Atom
- ▯ Server: localhost server
- ▯ Operating System: Windows 10

Implementation technology:

- ▯ Programming language: PHP
- ▯ Pattern: Web API 2
- ▯ Framework: Laravel framework 5.8.10

CHAPTER 5 SYSTEM TESTING

5.1 Testing Features:

Testing is one of the most important parts of checking the functionality of a software or system. Testing every feature of software we can ensure that the software or system is ready to provide its service [6].

5.1.1 Features to be tested:

There is given a list of the features we have tested

- Admin Login
- Owner message
- Customer message
- Owner sends data
- Buy and sell dog

5.1.2 Features not to be tested:

We have tested every field of our system.

5.2 Testing strategies:

There is given the strategies of our system or software testing.

- Quality test
- Measure test

5.2.1 Test Approach:

To implement a project test approach is very important. Test approach has two types of techniques:

- **Proactive** - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.
- **Reactive** - An approach in which the testing is not started until after design and coding is completed.

5.2.2 Pass/Fail criteria:

- When the expected result is come after testing then it is called pass
- When an unexpected result comes during testing then it is called fail.

5.2.3 Suspension & Resumption:

Our system is very smooth and easy to use and maintain. There is no suspension and resumption problem.

5.3 Testing Environment:

- Testing IDE : PhpStrom
- Browser : Google Chrome
- Server : Local host server
- Operating System: Windows 10
- Possessor : Core i3

5.4 Test Case Description:

We have tested our system many times. Now we are focusing on that our system will execute properly.

5.4.1 Test Case for Admin Login:

Table 5.1: Test Case for Admin Login

Test Scenario:	Admin Login	Test Case ID:	TC01		
Test Case Description:	Login test case	Test Priority:	High		
Pre-Requisite:	A valid user account	Post-Requisite:	NA		
Test Execution Steps:					
SL. No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Enter correct Email and password	Email: saif@gmail.com Password: *****	Login Successful	Login Successful	Pass

5.4.2 Test Case for verifying owner:

Table 5.2: Test Case to verifying the owner

Test Scenario:	Verify owner	Test Case Id:	TC02		
Test Case Description:	Verify owner Test Case	Test Priority:	High		
Pre-Requisite:	An Authenticated Admin	Post-Requisite:	Na		
Test Execution Steps:					
SL. No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Enter Correct Dog's info message Send By owner	Dog name: Jack Price: Negotiable Owner no: 017XXXXX	Owner message Successful	Update States Successful	Pass

5.4.3 Test Case for customer message:

Table 5.3: Test Case for customer message

Test Scenario:	Customer message	Test Case ID:	TC03		
Test Case Description:	Message test case	Test Priority:	High		
Pre-Requisite:	A valid message	Post-Requisite:	NA		
Test Execution Steps:					
SL. No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Enter Name send message	Name: Rahim Message: Please response my mail	message Successful	send Successful	Pass

5.4.4 Test Case for customer buy dog:

Table 5.4: Test Case for customer buy dog

Test Scenario:	Customer buy dog	Test Case ID:	TC04		
Test Case Description:	Customer buy dog	Test Priority:	High		
Pre-Requisite:	An authenticated owner	Post-Requisite:	NA		
Test Execution Steps:					
SL. No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Enter correct name Of customer	Name: Ashik	Deal Successful	Deal Successful	Pass

5.4.5 Test Case for owner sell dog:

Table 5.5: Test Case for owner sell dog

Test Scenario:	Owner sell dog	Test Case ID:	TC05
Test Case Description:	Owner sells dog test case	Test Priority:	High
Pre-Requisite:	An authenticated customer	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Negotiate Price With customer	All required information	Registering marriage Successful	Registering marriage Successful	Pass

CHAPTER 6

USER MANUAL

6.1 Home Page: This is our system's home page. Through the home page owner and the customer can see the project easily. And also find all the information they needed. User can find their dog on the home page. There is many dog's info on the home page.

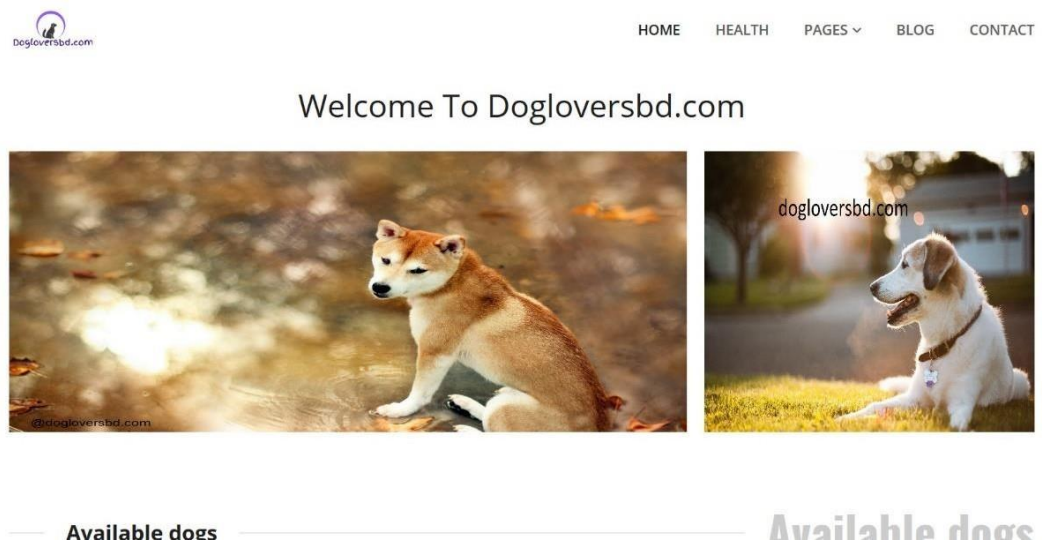


Figure 6.1: home page



Figure 6.2: home page



Figure 6.3: home page

6.2 dog's info: When a user chooses a dog. Then click on the dog's pic. then the user can see the dog's info. Also, see the dog's owner phone number and the dog's breed name.

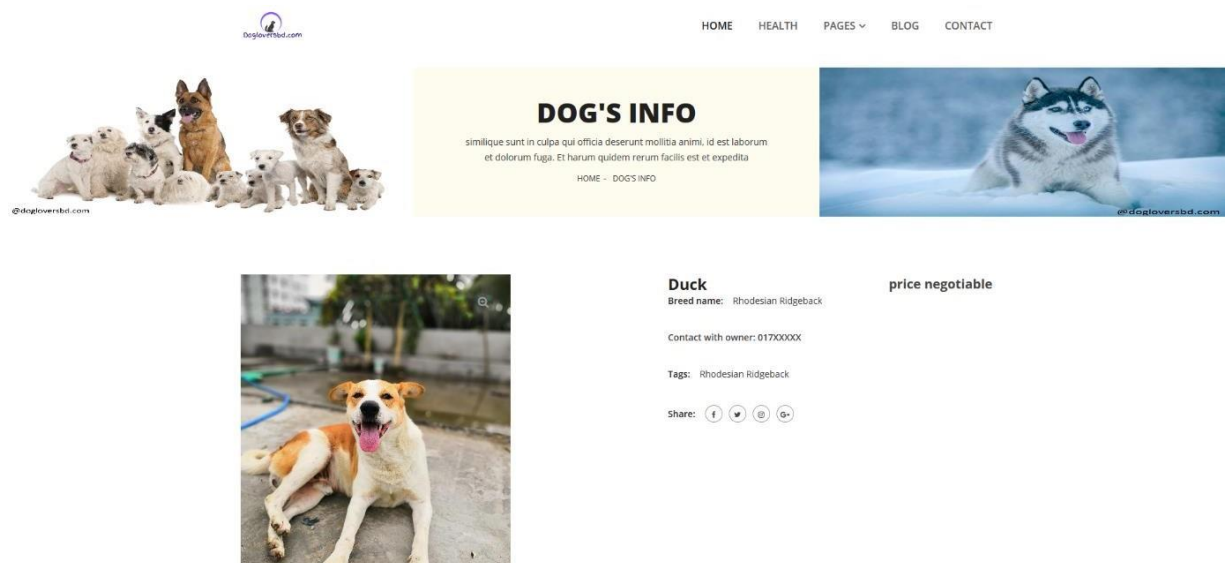


Figure 6.4: dog's info page

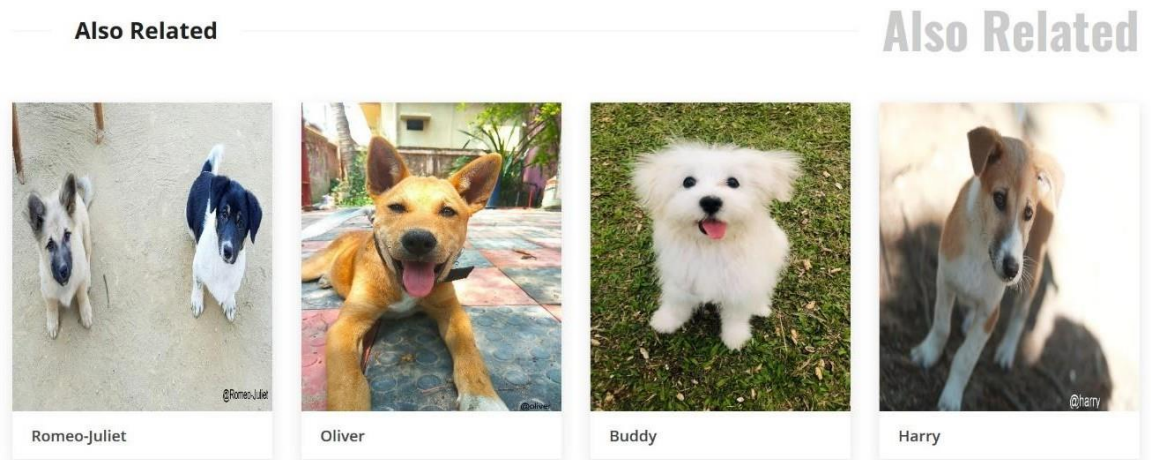


Figure 6.5: dog's info page

6.3 health page: My project also contains the health page. That might help the dog lovers to taken care of their dog.

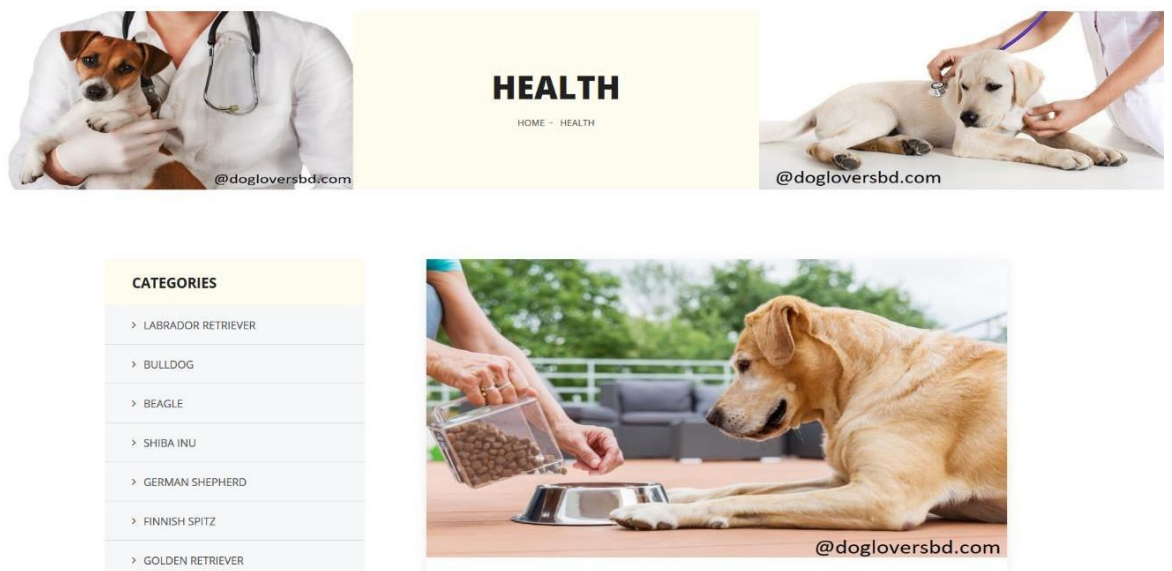


Figure 6.6: dog's health page



cleaning



play with them



Figure 6.7: health page

take to the veterinary sergen



please for more update like our facebook page

Figure 6.8: health page

6.4 Contact with the Facebook page: My project also provides the Facebook page of dog lovers group. So, might the owner and customer can contact with each other on Facebook.

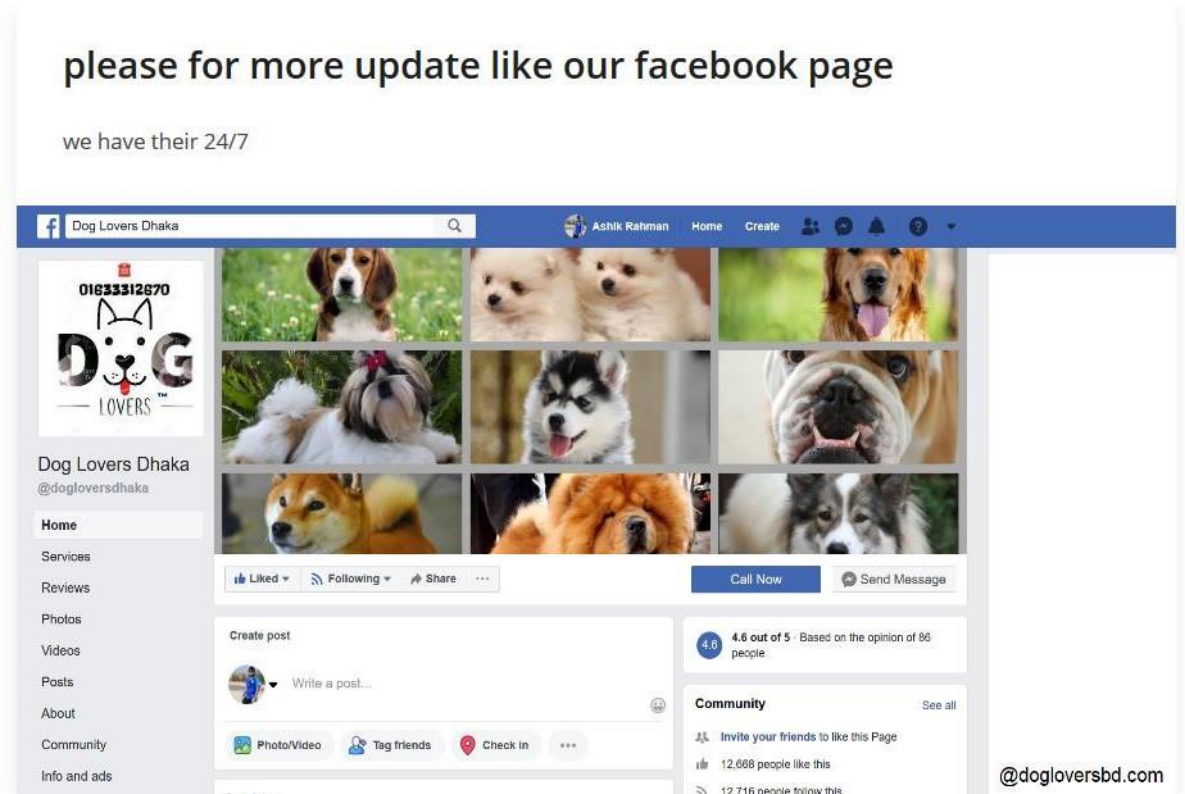


Figure 6.9: health page

6.5 contact page: In the contact page, the owner can message the admin.

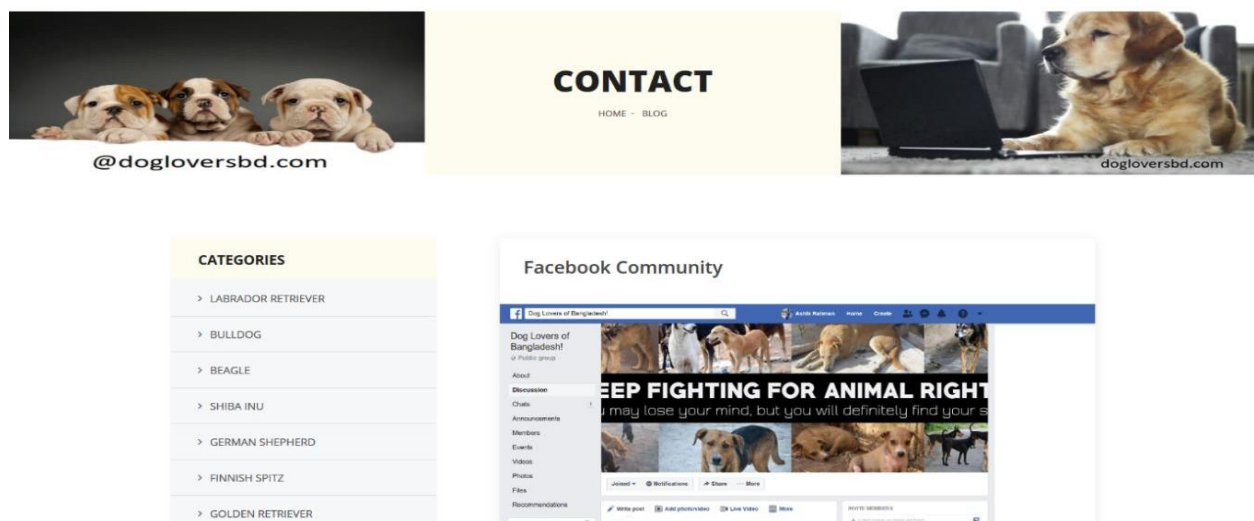


Figure 6.10: contact page.

Inform Us:

Enter your name

Enter email address

Enter phone

Enter message

Send Message

Figure 6.11: contact page

6.6 admin login: admin must log in to the project. And also see the dog's owner message. Admin reply to them.

Laravel

Login Register

Login

E-Mail Address ashikk521@gmail.com

Password ••••••••

☐ Remember Me

Login Forgot Your Password?

Figure 6.12 admin login

AdminPress

ashik

Main Navigation

- Dashboard
- Users
- Live site
- Logout

Dashboard

Home > Dashboard

All Contact Message

Name	Email	Phone	Message	Time	Manage
tonmoy	tonmoy@gmail.com	01887766	confirm my email	2019-04-11 15:06:09	+ -
ashik	ashik@gmail.com	01759343663	plz varify email	2019-04-03 17:35:35	+ -
sumon	saha@gmail.com	017338596	plz add some golden...	2019-03-24 16:27:47	+ -
fahad	fahad@gmail.com	01761224010	i want to see some n...	2019-03-24 16:20:14	+ -
ashikkk	ashikkk@gmail.com	567675568	hi post some new	2019-03-24 15:34:58	+ -
plabon	plabon@gmail.com	37739459645	response my email	2019-03-24 15:34:26	+ -
biplob	biplob@gmail.com	019763	plz email me i want...	2019-03-24 07:09:34	+ -

Figure 6.13: admin contact list

AdminPress

ashik

Main Navigation

- Dashboard
- Users
- Live site
- Logout

Dashboard

Home > Dashboard

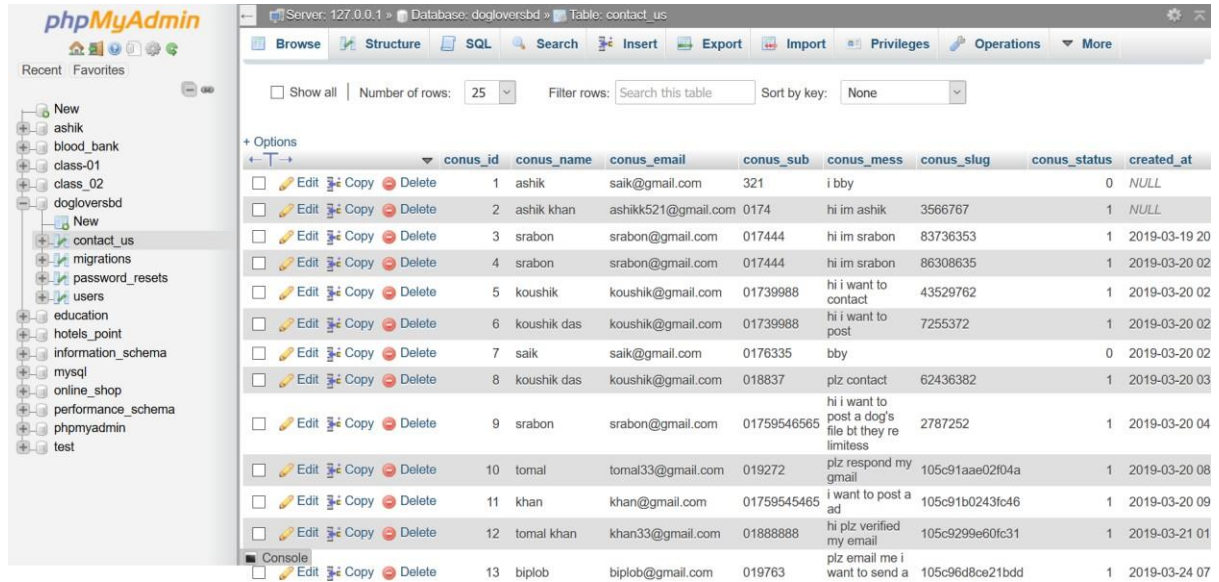
All Contact Message

Name	: tonmoy
Email	: tonmoy@gmail.com
Phone	: 01887766
Message	: confirm my email
Time	: 2019-04-11 15:06:09

Excel

Figure 6.14: message list details

6.7 localhost details: admin access to the local host. And see the user.



Server: 127.0.0.1 • Database: dogloversdb • Table: contact_us

Options: Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	conus_id	conus_name	conus_email	conus_sub	conus_mess	conus_slug	conus_status	created_at
☐ Edit <input type="copy"/> <input type="delete"/>	1	ashik	saik@gmail.com	321	i bby		0	NULL
☐ Edit <input type="copy"/> <input type="delete"/>	2	ashik khan	ashikk521@gmail.com	0174	hi im ashik	3566767	1	NULL
☐ Edit <input type="copy"/> <input type="delete"/>	3	srabon	srabon@gmail.com	017444	hi im srabon	83736353	1	2019-03-19 20
☐ Edit <input type="copy"/> <input type="delete"/>	4	srabon	srabon@gmail.com	017444	hi im srabon	86308635	1	2019-03-20 02
☐ Edit <input type="copy"/> <input type="delete"/>	5	koushik	koushik@gmail.com	01739988	hi i want to contact	43529762	1	2019-03-20 02
☐ Edit <input type="copy"/> <input type="delete"/>	6	koushik das	koushik@gmail.com	01739988	hi i want to post	7255372	1	2019-03-20 02
☐ Edit <input type="copy"/> <input type="delete"/>	7	saik	saik@gmail.com	0176335	bby		0	2019-03-20 02
☐ Edit <input type="copy"/> <input type="delete"/>	8	koushik das	koushik@gmail.com	018837	plz contact	62436382	1	2019-03-20 03
☐ Edit <input type="copy"/> <input type="delete"/>	9	srabon	srabon@gmail.com	01759546565	hi i want to post a dog's file bt they re limitless	2787252	1	2019-03-20 04
☐ Edit <input type="copy"/> <input type="delete"/>	10	tomal	tomal33@gmail.com	019272	plz respond my gmail	105c91aae02f04a	1	2019-03-20 08
☐ Edit <input type="copy"/> <input type="delete"/>	11	khan	khan@gmail.com	01759545465	i want to post a ad	105c91b0243fc46	1	2019-03-20 09
☐ Edit <input type="copy"/> <input type="delete"/>	12	tomal khan	khan33@gmail.com	01888888	hi plz verified my email	105c9299e60fc31	1	2019-03-21 01
☐ Edit <input type="copy"/> <input type="delete"/>	13	biplob	biplob@gmail.com	019763	plz email me i want to send a	105c96d8ce21bdd	1	2019-03-24 07

Figure 6.15: phpMyAdmin local host details

6.8 breed list details:

CATEGORIES

- > LABRADOR RETRIEVER
- > BULLDOG
- > BEAGLE
- > SHIBA INU
- > GERMAN SHEPHERD
- > FINNISH SPITZ
- > GOLDEN RETRIEVER
- > RHODESIAN RIDGEBACK

Types of dogs breed

Labrador Retriever

The Labrador Retriever, or just Labrador, is a large type of retriever-gun dog. The Labrador is one of the most popular breeds of dog in Canada, the United Kingdom and the United States.




Figure 6.16: breed list details

CHAPTER 7

PROJECT SUMMARY

7.1 GitHub Link:

Project upload in GitHub. [8]

7.2 Critical Evolution:

Requirement engineering was a challenge for me because many owners were not willing to share information. My system is not available for all religions until now.

7.3 Limitations:

Every system has some limitations. Because a system is not built for all services or not for all over the world. Like that my system has some limitations given below.

- ▯ My project is based on Bangladeshi perspective, rules, and regulations.
- ▯ My project is only to taking message and response owner.
- ▯ This is a beta version.

7.4 Obstacles & Achievements:

I have to take the challenge to think or do something new. Because I have to face some difficulties as obstacles to do something new. When I can complete the new thing then I can learn something new and can overcome the obstacles then these are my achievements.

- ▯ Collecting requirements was a great challenge and obstacle.
- ▯ Many people are not aware of the internet in our country.
- ▯ After completing this project, I have learned many things.

7.5 Future Scope:

I am trying to make an android app for my users. In the future, I am trying to add other pets beside the dog. Made this project huge and release it internationally. Digital payment the system can be adding in the future.

7.6 Discussion and Conclusion

Nowadays maintain security in an area is a big problem. To maintain security the admin should have the information about the system. My app is implemented to maintain information about people. So, it will render a great effort to the owner and the customer. In the age of computer science, an analog system for maintaining information is an old process. People can serve their personal information by using their android phone. It will also save time. Using my app owner can easily maintain all the information at a time in a single database.

If the user fills free to use my app and accept my work in a cordial heart then the implementation of the application will succeed.

REFERENCES

- [1]. (Block Diagram, 31 July,2008)
https://en.wikipedia.org/wiki/Block_diagram, last accessed: Jan 10, 2019.
- [2]. (Gantt chart, 1910)
https://en.wikipedia.org/wiki/Gantt_chart, last accessed: Apr 20, 2019.
- [3]. (Activity Diagram, 20 July, 2008) https://en.wikipedia.org/wiki/Activity_diagram,
last accessed: Feb 4, 2019.
- [4]. UML Sequence Diagram Tutorial, <https://www.smartdraw.com/uml-diagram/>
last accessed: Feb 22, 2019.
- [5]. Database design. https://en.wikipedia.org/wiki/Database_design.
Last accessed: Mar 15, 2019.
- [6]. Microsoft SQL Server,
<https://www.microsoft.com/en-us/sql-server/application-development>
last accessed: Mar 25, 2019.
- [7]. Learn UI Design, <https://learnui.design/>
last accessed: Apr 2, 2019.
- [8]. <https://github.com/srabon77/dogloversbd.com>. Last accessed: Apr 3, 2019.