A picture containing dark

Description automatically generated

**DOG CARE**

**Development Of An Application To Support The Management Of Chronic Diseases In Dogs**

by

**PARDEEP KUMAR**

This dissertation was submitted in part fulfilment of requirements for the degree of MSc Advanced Software Engineering.

Logo, company name

Description automatically generated

Department of computer and information sciences

August 2022

Text, application, letter

Description automatically generated

**ABSTRACT**

Text, letter

Description automatically generated

**ACKNOWLEDGEMENT**

I thank Dr Roma Maguire of the University of Strathclyde for supervising this work.

Table of Contents

[Table of Figures x](#_Toc111812699)

[1. Introduction 1](#_Toc111812700)

[1.1 Background of the problem 1](#_Toc111812701)

[1.2 Aims and Objectives 2](#_Toc111812702)

[2: Literature Review 3](#_Toc111812703)

[2.1 Diseases in Dogs and their symptoms 3](#_Toc111812704)

[2.1.1 Chronic diseases 3](#_Toc111812705)

[2.1.2 Canine distemper 4](#_Toc111812706)

[2.1.3 Canine parvovirus (CPV) 4](#_Toc111812707)

[2.1.4 Osteoarthritis (OA) 5](#_Toc111812708)

[2.1.5 Concluded symptoms 5](#_Toc111812709)

[2.2 Existing research work 6](#_Toc111812710)

[2.3 Comparative market analysis of relevant applications 8](#_Toc111812711)

[2.3.1 FirstVet 8](#_Toc111812712)

[2.3.2 11pets 9](#_Toc111812713)

[2.3.3 PetDialog 10](#_Toc111812714)

[2.3.4 Dog Buddy 11](#_Toc111812715)

[2.3.4 Concluded requirements/features 11](#_Toc111812716)

[2.4 DOG CARE: Why to develop? 12](#_Toc111812717)

[3: Methodology and Requirements Gathering 13](#_Toc111812718)

[3.1 Agile methodology 13](#_Toc111812719)

[3.2 Requirements Gathering 15](#_Toc111812720)

[3.2.1 Selection of requirement gathering technique 16](#_Toc111812721)

[3.2.2 Interviews 17](#_Toc111812722)

[3.2.3 User Personas 20](#_Toc111812723)

[3.2.4 Use Case Diagram 22](#_Toc111812724)

[3.2.4 Functional and Non-Functional Requirements 23](#_Toc111812725)

[4. Design 28](#_Toc111812726)

[4.1 System Architecture 28](#_Toc111812727)

[VIPER Architecture 29](#_Toc111812728)

[4.2 User Interface Design 30](#_Toc111812729)

[4.2.1 Low-Fidelity Sketches 31](#_Toc111812730)

[4.2.2 Mid Fidelity Wireframes – Dog Owners 32](#_Toc111812731)

[4.2.3 Mid Fidelity Wireframes – Vet role 33](#_Toc111812732)

[4.3 Data Design 34](#_Toc111812733)

[4.3.1 Flowchart: 35](#_Toc111812734)

[4.4 High Level Design 36](#_Toc111812735)

[4.5 Technology and Languages 36](#_Toc111812736)

[4.4.1 Flutter 37](#_Toc111812737)

[4.4.2 DART 38](#_Toc111812738)

[4.4.3 PHP (Backend) 39](#_Toc111812739)

[5. Implementation 41](#_Toc111812740)

[5.1. Build 41](#_Toc111812741)

[5.1.1 Login/Registration screen 43](#_Toc111812742)

[5.1.2 Reset Password 44](#_Toc111812743)

[5.1.3 View/Update Profile 45](#_Toc111812744)

[5.1.4 Home Screen 46](#_Toc111812745)

[5.1.5  Create Pet 47](#_Toc111812746)

[5.1.6 Dog Details 48](#_Toc111812747)

[5.1.7 Symptom Tracker Screen 49](#_Toc111812748)

[5.1.8 Conclusion Screen 50](#_Toc111812749)

[5.1.8 Report 51](#_Toc111812750)

[5.1.9 Request Screen 51](#_Toc111812751)

[5.1.10 Settings screen – Pet Owner 52](#_Toc111812752)

[5.1.11 Symptom Tracker History 54](#_Toc111812753)

[5.1.12 All Request Screen – Vet Client 54](#_Toc111812754)

[5.1.13 Settings Screen – Vet Client 55](#_Toc111812755)

[5.1.14 User Journey – User / Dog owner Application 57](#_Toc111812756)

[5.1.15 : User Journey – Vet Client App 59](#_Toc111812757)

[5.2 Life Cycle and project Management 60](#_Toc111812758)

[5.3 Coding Standards 61](#_Toc111812759)

[5.3.1 Naming Conventions 62](#_Toc111812760)

[5.3.2 Class Ordering 62](#_Toc111812761)

[5.3.3 Formatting Guidelines 63](#_Toc111812762)

[5.3.4 Comments 63](#_Toc111812763)

[5.4 Sprints 63](#_Toc111812764)

[6. Testing and evaluation 65](#_Toc111812765)

[6.1 Usability 65](#_Toc111812766)

[6.2 Heuristic Evaluation 66](#_Toc111812767)

[6.3 Mobile testing 68](#_Toc111812768)

[6.4 Unit testing 68](#_Toc111812769)

[6.5 System testing 69](#_Toc111812770)

[6.6 User acceptance testing 69](#_Toc111812771)

[6.6.1 Application Deployment 69](#_Toc111812772)

[6.6.2 Users feedback and product refinement 69](#_Toc111812773)

[7. Conclusion and Future Work 74](#_Toc111812774)

[7.1 Conclusion 74](#_Toc111812775)

[7.2 Future Work 75](#_Toc111812776)

[7.2.1 Multiple Diseases Functionality 75](#_Toc111812777)

[7.2.2 Registration of vet from admin panel 75](#_Toc111812778)

[7.2.3 Group Chat Functionality 75](#_Toc111812779)

[7.2.4 Notifications feature 76](#_Toc111812780)

[7.2.5 Real time chat with vet 76](#_Toc111812781)

[7.2.6 Freedom to choose vet 76](#_Toc111812782)

[7.2.7 User Interface Enhancements 76](#_Toc111812783)

[7.2.7 Deployment of application on AppStore and Play Store 77](#_Toc111812784)

[References 78](#_Toc111812785)

[Appendices 84](#_Toc111812786)

# Table of Figures

[Figure 1: FirstVet application screenshots 8](#_Toc111812787)

[Figure 2: 11pets application screenshots 9](#_Toc111812788)

[Figure 3: PetDialog application screenshots 10](#_Toc111812789)

[Figure 4: Dog Buddy application screenshot 11](#_Toc111812790)

[Figure 5: Agile methodology 14](#_Toc111812791)

[Figure 6: Pet owner Persona 21](#_Toc111812792)

[Figure 7: Vet Persona 21](#_Toc111812793)

[Figure 8: Use Case Diagram 22](#_Toc111812794)

[Figure 9: Functional requirements 25](#_Toc111812795)

[Figure 10: Non-Functional requirements 27](#_Toc111812796)

[Figure 11: System Architecture 29](#_Toc111812797)

[Figure 12: Viper Architecture 30](#_Toc111812798)

[Figure 13: Low fidelity sketch of application 31](#_Toc111812799)

[Figure 14: Low fidelity sketches of different screens of application 32](#_Toc111812800)

[Figure 15: Detail Screen, Symptom Tracker, Conclude Screen 33](#_Toc111812801)

[Figure 16: Dashboard Vet App, Report Screen(Replied), Report Screen(Pending) 34](#_Toc111812802)

[Figure 17: Flowchart of application 35](#_Toc111812803)

[Figure 18: Database Schema 42](#_Toc111812804)

[Figure 19: Login / Registration 43](#_Toc111812805)

[Figure 20: Reset Password 44](#_Toc111812806)

[Figure 21: View/Update Profile 45](#_Toc111812807)

[Figure 22: Home Screen 46](#_Toc111812808)

[Figure 23: Create Pet 47](#_Toc111812809)

[Figure 24: Dog Details 48](#_Toc111812810)

[Figure 25: Symptom Tracker 49](#_Toc111812811)

[Figure 26: Conclusion Screen 50](#_Toc111812812)

[Figure 27: Report 51](#_Toc111812813)

[Figure 28: Request 52](#_Toc111812814)

[Figure 29: Settings 53](#_Toc111812815)

[Figure 30: Symptom Tracker History 54](#_Toc111812816)

[Figure 31: All Requests - Vet Client 55](#_Toc111812817)

[Figure 32: Settings - Vet Client 56](#_Toc111812818)

[Figure 33: User Journey - Dog Owner App 57](#_Toc111812819)

[Figure 34: User Journey - Vet Client 59](#_Toc111812820)

[Figure 35: Life Cycle of application 60](#_Toc111812821)

[Figure 36:Expert List Updated Design 70](#_Toc111812822)

[Figure 37:User Profile Updated Design 71](#_Toc111812823)

[Figure 38:Settings Updated Design 72](#_Toc111812824)

[Figure 39:Symptom Tracker Updated Design 73](#_Toc111812825)

# 1. Introduction

## 1.1 Background of the problem

Dogs are among the oldest domesticated animal species. The earliest evidence of dog domestication dates back over 9500 years ago in western Asia. Claiming a pet is a significant responsibility that will endure throughout the animal’s lifetime, maybe more than 10 years on account of dogs (Robinson, Lundgren and Segal, 2019). Dogs require their owners to take good care of them. Owners can easily be aware of the external issues of their pets but they get unaware of the internal conditions like chronic problems which may be leading towards a disaster for the pet owners. A dog's chronic condition also depends on proper care, maintenance and veterinary treatment. It is a common trend to neglect these serious issues as they are assumed to be normal physical condition. Even though owners never see these health issues being visible or having long-term effects like obesity, they may be causing a lot of other damage in your dog’s body. Dog obesity can lead to arthritis pain, back problems and heart failure due to excess weight pressing the insides of their joints. Obesity is a common chronic disease, which can also cause nerve damage which eventually leads to fatigue or even death for some dogs (Varshneya, 2022). Osteoarthritis is a common and painful condition when the cartilage in dog’s joints deteriorates due to age, injury, or previous joint surgeries. If left untreated, osteoarthritis can lead to chronic pain and loss of function. So it would be best if such disease is diagnosed in early stages. So, chronic disease management has become a matter of concern in animal health care, which has led to a rise in the adoption of technological innovations for better managing animal health.

Moreover, in a period where virtually every customer consumable products such as books, food etc. can be bought through a mobile application on a cell phone, it appears to be sensible to imagine that increasingly more of our medical services can likewise be managed utilizing applications on cell phones. Development of these applications see the capability of advanced innovations to move attention from doctor’s clinic and emergency health care to the patient's home or anyplace with a suitable Wi-Fi network. And while there are plenty of existing mobile application on the marketplace for caring about dog’s health, but they don’t fulfil all the user requirements like tracking the symptoms of chronic diseases without visiting the vet. Sometimes, pet owners have to visit the vet just for checking mild symptoms as they don’t have any system that can conclude when they need to visit the vet. This begs the requirement of an application which provide users a support to track early, mild and severe symptoms of chronic disease while providing the benefits of consultation without visiting the vet?

## 1.2 Aims and Objectives

Dog Careis a mobile application that aims to serve the following functions:

* To provide different user roles for pet owners and vets.
* To track the symptoms of chronic disease (Arthritis) in dogs
* To provide remote consulting by analysing symptoms.
* To raise awareness among the dog owners and pet lovers.

This application will play an important role for the pet owners to get medical assistance for their pets and it will likewise assist them with understanding that their pets are in the grip of a few illnesses. This application will help them in deciding when they should take their pet to the vet as this app focuses on tracking the severity of the disease by categorizing the symptoms. On the other hand, this application will also help the vets to provide feedback on pet health by analysing the report of symptoms that will be tracked by the application. This application can follow changes in pet like the way of behaving and development, recognizing changes which could be something to worry about. This permits users to explore and identify early indications of sickness, which is frequently significant to treat conditions and guarantee dog’s well-being (Kamishina et al. 2019).

# 2: Literature Review

## 2.1 Diseases in Dogs and their symptoms

Dogs can suffer from several chronic conditions and numerous behavioural problems that may or may not be able to be helped with traditional pet medications. If the dog has suddenly stopped eating his food or is exhibiting strange behaviour this may be a sign that he needs to see the vet for an examination. A dog's health and well-being can be affected if they don't have proper care, regular maintenance, or veterinary treatment. Although it is impossible to have a healthy pet without health problems, it is also common for owners to think that their dog's specific illness is normal and not cause a lot of damage to their pets' body. For example, obesity can be caused by neglecting proper nutrition for your dog. Plus, even though a dog may not show any symptoms, chronic conditions like epilepsy can result in major long-term health issues. Chronic conditions that persist for a long period of time and which cannot be effectively managed with usual methods may require complementary or alternative treatment, comprising the use of conventional treatments and complementary therapies.

Dogs are susceptible to a variety of conditions that range from mild to serious and life-threatening. Some dogs will develop diseases that affect the quality of their life in several undesirable ways, including being isolated from other dogs, having trouble bonding with their owner, or becoming aggressive toward family members if they cannot gain access to food and water. Dogs have diseases of every organ and tissue, including their skin and musculoskeletal systems. In addition to physical disease, many behavioural issues because of owners' great stress, hasten the decline in the quality of life of their pet and lead to early death (Kani et al. 2019). Some of the major diseases of dogs are as follows:

### 2.1.1 Chronic diseases

Chronic diseases are those health conditions that can last for months or longer. They are typically caused by many factors, including old age, lifestyle choices, genetics, and environmental factors. Chronic conditions often affect older dogs, but they can affect dogs of any age; however, they can occur in younger dogs as well with the right conditions such as obesity and high-stress levels. Many health problems go away after the dog is treated, but that isn’t always the case. Changes in diet and activity levels may be needed to keep your dog comfortable and healthy (Kim et al. 2019). All dogs can develop diseases and disorders, but some are more common in overweight and older dogs than others. The American Society of Veterinary Nutrition states that older dogs with chronic conditions have a 35% chance of developing osteoarthritis (OA) in both elbows, a condition that causes moderate to severe pain, stiffness and immobility (Christian et al. 2018). Dogs can also develop other conditions as detailed below.

### 2.1.2 Canine distemper

Although the signs of Canine distemper in dogs are like those in wild canines, it is important to look out for symptoms that may be more severe such as diarrhoea, coughing, and decreased appetite. These might also be accompanied by fever. The initial stages of canine distemper involve a dog developing watery to pus-like discharge from their eyes. Later, the dog develops a fever, nasal discharge, coughing, and lethargy in addition to reduced appetite and vomiting. The initial stages of canine distemper involve a dog developing watery to pus-like discharge from their eyes. Later, the dog develops a fever, nasal discharge, coughing, and lethargy in addition to reduced appetite and vomiting. Canine distemper is a viral infection that causes a wide range of signs in dogs. Although the signs of canine distemper are similar to those in wild canines, it is important to look out for symptoms that may be more severe such as diarrhoea, coughing, and increased appetite. These might also be accompanied by fever and haemorrhage at the sites of injection (Christian et al. 2018).

### 2.1.3 Canine parvovirus (CPV)

Canine parvovirus (CPV) is highly contagious and is spread from dog to dog by direct or indirect contact with their feces, saliva, and respiratory secretions. High environmental hygiene plays an important role in reducing the transmission rate of this virus among dogs. There are no apparent complications that arise from infection with the parvovirus enteritis viruses, however, both acute and chronic infections may occur, especially if infected dogs are housed in crowded environments for extended periods. Dogs that are confined to a building can transmit CPV to other dogs. In addition to accidental transmission, it is recommended that pet owners make sure that their pets do not come into contact with other dogs or humans when exhibiting signs of illness (Márquez, 2018). Canine parvovirus (CPV) is an infectious viral disease that affects canines and other warm-blooded mammals, including humans. It causes a high fever, depression, and unusual loss of appetite that may last for weeks. CPV can be spread between canines through direct contact with infected urine or feces, or indirectly when another animal's stool contaminates the environment around them. There is no cure for CPV in dogs; medical treatment focuses on alleviating symptoms, allowing time for the body to heal itself (Márquez, 2018).

### 2.1.4 Osteoarthritis (OA)

Osteoarthritis is a common and painful condition when the cartilage in a dog's joints deteriorates due to age, injury, or previous joint surgeries. If left untreated, osteoarthritis can lead to chronic pain and loss of function. Frequent exercise and joint-supporting supplements can help keep your dog comfortable and active throughout its golden years! Other symptoms of osteoarthritis include stiff, swollen or sore joints Reluctance to be touched on some parts of the body (particularly after exercise) Loss of stamina, unexpected aggression toward other dogs or humans: A veterinarian can help diagnose and treat a dog's osteoarthritis. The best way to prevent osteoarthritis is to keep your dog active and fit. Suppression of appetite Loss of stamina Unexpected aggression toward other dogs or towards humans Stiff joints A decrease in activity levels Despite their age, many senior dogs can still be active and enjoy life. However, as the dog ages, it’s important to pay attention to their body language and behaviour changes.

### 2.1.5 Concluded symptoms

Based on the above diseases, the following symptoms were identified to be included in the application to track the severity of disease:

* Difficulty getting up and down
* Walking stiffly
* Lameness in one or more legs
* Reluctance to go up and/or downstairs
* Reluctance to jump up or down (onto/off furniture or into/out of a vehicle)
* Stiff, swollen, or sore joints
* Reluctance to be touched on some parts of the body
* Loss of stamina
* Unexpected aggression towards other dogs or towards humans

## 2.2 Existing research work

Considering the increasing adoption of advanced applications appearing to fulfil contemporary personal health demands, it's crucial to comprehend how well these applications can be utilized for a variety of objectives. Several mobile applications check cardiac rehabilitation and overall health indicators, and others can measure aspects such as sleeping habits, using multiple alerting systems to get up at the appropriate time, and so forth.

Since the healthcare business is expanding so quickly, mobile applications are indeed being used to record healthcare information. Remote monitoring technologies are used in a variety of healthcare activities. There are mobile applications for rehabilitation, such as "MARHS," food modification, such as "Myca Nutrition," "biomarkers" for patients with heart failure as well as other overall health measures, and so on (Ana et al. 2020). All these programs are for humans, but there are relatively few applications available for tracking the location of pets but not their health. One of the research work monitors the canine activity in dogs. (WagTag). But this work does not provide the virtual medical assistance.

This initiative resulted in the creation of a smartphone health app for dogs. This dissertation project's app is unique in that it is focused on mobile devices rather than electrical equipment. "It tracks dog healthcare-related conditions using intelligent devices and cellphones, and this data is subsequently used by vets to evaluate the dog's condition." Veterinarians employ an advanced "Electronic Medical Record" (EMR) program to easily obtain a dog's medication history, medical issues, and any updated data ever since the dog's prior consultation (Berihun et al. 2020). This health system can also enable them to interact directly with each animal owner about the general health of their pets.

This mobile application's key features are information tracking and analytics. This program can assist the veterinarian in reviewing the dog's documentation prior to the inspection, updating the dog's background throughout a session, and refilling digital medications. The thesis work is based on the idea of collecting dog information and using analytical and visualization methods to assist vets understand better their patients' requirements. It does, however, utilize the dog's numerous data points to monitor their healthcare regimen. It is advantageous to veterinarians since they can now monitor their patients and drugs in real time. By combining many datasets into a unified platform, professional veterinarians who may never have sufficient time to handle anything manually gain extra time. This program helps the vet's operations from anywhere and anytime. This mobile application is a cloud-based web platform that connects veterinarians to the owner in real time and offers multiple features for the canine health care plan. Information tracking and analytics are key features of this program. The application will assist vets in reviewing their dog's documentation prior to an inspection, updating the dog's background throughout a session, and refilling digital medications. The program helps veterinarians carry out their duties more efficiently, especially in remote areas where they may not have sufficient time to manage things manually (Guard et al. 2019).

This application, like "Timestation and EZ Office Inventory”, scans the dog's prescription data with a QR code and uploads it to cloud servers, where it can be accessed at any time and from any location (Bhanote et al. 2022). As a result, consumers may check their dog's prescription information at any time from any location. The web record can be preserved, duplicated, and redistributed. The application also keeps an online database with the most recent acquisitions and images of the dog to keep you updated on its condition. There has been no infrastructure to maintain because it operates in the cloud.

## 2.3 Comparative market analysis of relevant applications

As we know that most of the work is digitized these days in almost every sector, the same is in the health sector. We can find many mobile applications related to dog health. For this research, some applications were analysed to understand the market requirement needed for this work. Below are some of the applications and their features with screenshots that are used for dog health analysis.

### 2.3.1 FirstVet

FirstVet is an independent digital veterinary application developed in 2021 that offers video consultations to pet owners for advice and guidance on their pets - wherever you are. This application provides following features:

* Consulting online/Video Consultation
* Booking appointment with vet
* Searching facts about animals
* Adding Pet Record

Some screenshots of this app are shown below:

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 1: FirstVet application screenshots

### 2.3.2 11pets

11pets is an independent digital veterinary application developed in 2021 that provides a complete medical record of the pet, whether a dog, cat, rabbit, bird, snake or any other species is crucial for treatment. This application provides following features regarding medical and care of dogs:

* Medical: Diagnosis and Tests, medications, surgeries, allergies, medical conditions, vet visits
* Care: hygiene, vaccination, cleaning, incidents, food, oestrus cycle, measurements
* Other Features: calendar, gallery, share report/documents

Below are some screenshots of this app:

Graphical user interface, application

Description automatically generated

Figure 2: 11pets application screenshots

### 2.3.3 PetDialog

PetDialog is a mobile application that was developed in 2017 which aims at recording a pet's daily routine, vaccinations and providing medical assistance. This application provides following features:

* Tracking Pet Activities: walking, food, habits, weight, etc
* Providing vaccination reminder
* Organizing calendar events
* Providing Vet Assistance

Screenshots of the app are provided in the below figure:

Graphical user interface, application

Description automatically generated

Figure 3: PetDialog application screenshots

### 2.3.4 Dog Buddy

Dog Buddy is a mobile application that was developed in 2011. This includes writing diaries, recording milestones and storing pictures as souvenirs, tracking dog's weight, saving dog pictures, and keeping track of dog's vaccinations, medications, and allergies. A user can keep track of milestones such as the birth of a dog, the growth of teeth, the fall in love of a dog, and so on.

This application provides following features:

* Adding Dog Detail: Medications, Vaccinations, Allergies, Weight, Others
* Getting Vaccine Reminders
* Adding More Pets

Dog buddy app screenshot is provided in following figure:

Graphical user interface, application

Description automatically generated

Figure 4: Dog Buddy application screenshot

### 2.3.4 Concluded requirements/features

After analysing all of the above applications, it was found that most of the existing application related to dog health provide following functionalities:

* Login/ Register
* Record of dog’s everyday routine, medications, vaccinations, walking & other details.
* Vet assistance based on data collected, video calling, booking appointments, chatting with vets.

Based on these features, some of the requirements were gathered to create my work.

## 2.4 DOG CARE: Why to develop?

As described in the earlier section, DOG CARE application aims to provide remote monitoring of symptoms of chronic diseases(Arthritis) and providing virtual assistance. The aim of this application was identified after considering the already available systems in the market and by understanding what users need from such a type of application which is not provided in the already available systems. Most of the applications provide functionalities like adding pet records and getting virtual assistance. But as suggested by one of the users, proper tracking of symptoms was not provided in any of the existing work. So, this was included as a requirement of this application. If symptoms are tracked properly, users would have to visit the vet only if symptoms match to the disease. Secondly, the vet can track disease with accurate symptom reports. Moreover, developing applications for pet owner and vet roles was also a feature lacking in existing applications. All these conditions lead to the development of the Dog Care application.

# 3: Methodology and Requirements Gathering

This section belongs to the methodology of this study. In this section, there is a discussion related to the flowchart of the application operation along with a detailed description of the application development phases. Similarly, the data collection process is also discussed in this section.

## 3.1 Agile methodology

As per the study of Beerbaum (2021), recently agile methodology is considered to be one of the most utilized practices in project management. It helps to provide enhanced business value, faster market, and greater transparency as well. Agile plays a significant role in mobile applications and Scrum is the subdomain that is used in the agile methodology. Agile is considered to be an interactive methodology in software development. It helps to organize the designs and planning the different methods, the phase of development and the various testing techniques during a complete software lifecycle. Agile methodology mainly has the basis on the various principles such as it provides quick response to all kinds of changes through adaptive planning. It provides the rationalization of different tasks which have to be performed. Since the current project is related to the application development, therefore agile methodology is integrated in the project. It includes the different stages such as the analysis of the requirements, design, development, testing, deployment and the review.

Further, only one cycle is not enough to create a full-fledged application, however each iteration shows the functionality part which can then be used to test or change. After each of the cycles, the development team checks all the requirements and makes changes which are needed to improve the result. Unlike the traditional desktop applications which can function for a lot of years without any upgrade, mobile applications should be made way more flexible. Agile methodology is helpful in developing the mobile application as it aids in providing in-depth planning in the real-time mode. This implies that it is extremely difficult to make a plan for the complete process of development (Hayat et al. 2019).

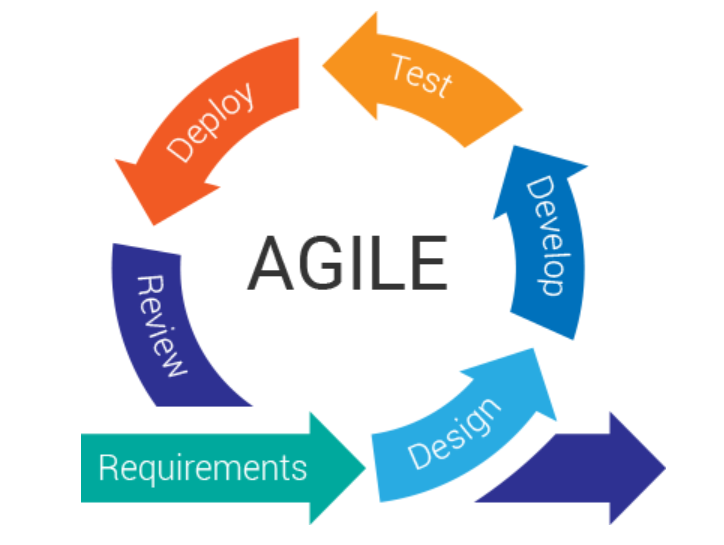


Figure 5: Agile methodology

Further, only one cycle is not enough to create a full-fledged application, however each iteration shows the functionality part which can then be used to test or change. After each of the cycles, the development team checks all the requirements and makes changes which are needed to improve the result. Unlike the traditional desktop applications which can function for a lot of years without any upgrade, mobile applications should be made way more flexible. Agile methodology is helpful in developing the mobile application as it aids in providing in-depth planning in the real-time mode. This implies that it is extremely difficult to make a plan for the complete process of development (Hayat et al. 2019).

With the help of agile methodology, a plan can be easily created for each of the cycles in a separate manner. This helps in saving the time and the number of resources used for fixing all the bugs. Thus, a proper plan can be created without any hindrance and aids in creating a first-rate product. Through agile, an application can be created sprint by sprint, cycle is referred to as the sprint. Further it is convenient to make quick changes in the mobile application as it is divided into sprints. As the current application requires continuous interaction from the customers, they can be helpful in providing feedback and the changes which are required in the application. Therefore, the agile methodology is the most suitable technique and used in the current scenario of dog application as well. A programme with many problems or poor functionality won't be used by users. It will result in the app's complete failure. Because of this, an app may be published incrementally using the agile technique, starting with a beta version that enables users to evaluate the app and report any defects they may encounter (Romero et al. 2021).

Developers can make all changes rapidly and handle all risks effectively based on it. When a consumer notices the final product of the development, it is not a scenario that can be tolerated. If a customer's expectations aren't met, it will be more challenging to make changes to the app, it will take more time and money, and it will result in adverse customer feedback. Agile technique enables the development team to communicate with customers continuously and deliver apps to them at the end of each sprint. In terms of modifying a product or a process, agile methodology is far more adaptable than waterfall. The agile technique makes it easy for teams to experiment and attempt something different during work if they feel the need to do (Abdelghany, Darwish and Hefni, 2019).

## 3.2 Requirements Gathering

Essentially, requirements describe the stakeholder's expectations of the software being developed, removing any ambiguity in terms of what will be built. As a result, stakeholders and developers must have a clear understanding of the requirements, which is documented in clear and precise terms. The nature of Agile methodologies is that they are continuously evolving. It is a continuous conversation between the client and the developer, and eventually between the client and the user, to gather requirements.

Diverging from traditional Agile projects the pet owner and the vet are the main stakeholders for the *Dog Care app*. A comparative market analysis was used to generate initial requirements, which were then finalized through the analysis of user interviews and finally adapted following user research and testing. Based on user research and testing, iterative development encourages the adaptation of requirements based on these functionalities. During the course of the development process, these initial requirements are likely to evolve.

### 3.2.1 Selection of requirement gathering technique

In order to ensure you don't have poor requirements, you should gather requirements properly and select the right technique. Many techniques are available for requirement gathering( **Mamoun Eid**, November, 2015). These are:

***a)One-on-one interviews :*** In addition to being one of the primary sources of requirements, one-on-one interviews are the most common method of gathering them. A well-prepared interviewer will help the interviewee get the most out of an interview. Primary data is information accumulated by experts from principal source materials such as interview sessions, experimental studies, and so on. It is derived from the source of information, where the majority of the information emanates (Lankani, 2021). As a result, it is the finest example of data gathering in research.

***b)Group interviews:*** The purpose of a group interview is similar to that of an individual interview, except it involves multiple people. These types of interviews give good results if the participants are at the same level.

***c)Questionnaire/Surveys:*** Analysers can collect information from a wide range of people via questionnaires or surveys in relatively short periods of time. But this is only a good option, when there are dozens or hundreds of respondents who need to be consulted to determine system requirements, or when stakeholders are geographically dispersed.

***d)JAD/AJAD:*** Joint Application Design(JAD) and Automated JAD are examples of more contemporary methods for gathering requirements.  As part of these techniques, information must be generated, analysed/decided upon, and documented.

**Selection of Technique**

Some initial requirements were gathered from the already existing systems in the market. But to understand what users need from a dog care application, the best way is to ask the users directly what they need. As this research was a single analyst work, so JAD or AJAD methods were not used for collecting requirements. Secondly, I did not have dozens or hundreds of people who could respond to my questions, so Survey method was also discarded and interviews were chosen. Some initial requirements were gathered from the already existing systems in the market as described in the literature review section. But to understand what users need from this application, the best way is to ask each individual directly what they require from a mobile application for managing the health of their pets. So, as per the user’ consent ,the interview method was chosen for gathering user requirements. The interviews are explained in detail in the below subsection.

### 3.2.2 Interviews

As already discussed earlier in this section, the motivation to do user interviews was to get the idea about the need of mobile application which will identify the arthritis in dogs at early stages.

For interviews, first of all, the users need to be identified which include those who interact with the current or new system, management, project financiers or anyone else that would be involved in the system. During interview preparation, it is necessary to ask general questions based on various individuals which require the interviewee to describe their views like asking about the current system. Some questions may require specific answers like true/false or multiple choice. After the questions have been decided, you should provide these questions to the interviewee before taking the interview so that they can prepare. User consent is necessary for recording the interview in order to save the details of the interview to ensure nothing is missed. Interview technique is very good in understanding user requirements. Some of the strengths and weaknesses of interviews are:

***Strength :***• Easy as they can be done with minimal preparation.  
• Interviews of one-on-one individuals require minimal planning effort as compared to that of a large group.  
• One-on-one interviews require less user commitment than large workshops.  
• One-on-one interviews provide a way to explore topics in detail.  
***Weakness :***• The interviewer's preconceived ideas may appear in the questions, influencing the interviewee's response.  
• The interview technique can be time-consuming and inefficient for projects with many stakeholders.  
• Additional interviews are needed to resolve conflicts and inconsistencies between stakeholder information.

**Interview Process:**

For conducting interviews, Microsoft Teams platform was chosen which provides excellent audio and video quality. First of all, an advertisement was placed in a local store and also online on social platforms like WhatsApp and Facebook to share with friends, colleagues and family. After that , consent of interested users was taken for taking audio interviews on Teams and recording the audio for saving information. After consent, invites were sent to the participants for an interview meet and finally, a 30 minutes duration interview was conducted with each participant. Two rounds of interviews were done with multiple dog owners, two men (35-40 years old) from Scotland, one 34 year old man from Germany, and one 36 years old man from Abu Dhabi. In the first round of interview, initial requirements were gathered and another round was done to get more suggestions and improvements along with scope of future implementations on the product developed so far. First rounds of interviews were started with brief introductory research, later on participants were asked to share their expectations related to the idea of application and use in real life. The questions for the second round of interview will be discussed in the evaluation section.

**Interview Questions:**

For gathering requirements, following questions were asked in the first round of interviews:

What features you want from a mobile application for arthritis in dogs.

What functions should the app have?

What content should be considered to develop an application like this.

Do you have such an application which can identify arthritis in dogs?.

Will you use an application to relieve the pain of their pet?

***Interview Conclusion:***

 After the first round of interviews, the following information was collected:

* Participant A explained that “I have a dog who is one and a half years old and I faced multiple issues related to my dog's health”. This app would help to provide a solution to his problems.
* He also suggested keeping dog related information such as pics, age information and other data related to previous recorded symptoms safe for future use.
* Moreover, Participant B suggested that “The app should have a feature of connecting with a vet”.
* While Participant C wanted to see other advanced AI base features like “dogs breed recognition as well as vet consultation”.
* Participant C also suggested having “notifications in application”. He suggested having multiple device compatibility and single handed use.
* Participant D mentioned that “I am using an iPhone but my other family members are using Android based smartphones so apps should be available for both trending technologies”.
* During interviews one dog owner suggested saving dog related data on cloud so that in future he can get details related to his dog quickly and easily to track if his dog is repeating the same symptoms again.

Based on the interviews, some user personas and user requirements were created which are explained in following subsections.

### 3.2.3 User Personas

Personas are hypothetical users that Alan Cooper, known as the "Father of Visual Basic," says “represent a certain group of users”. A persona is a made-up character with an interest in the system. The users' requirements, experiences, habits, and goals can all be understood on a single page through the personas. It also helps to identify the user of the product because it enables them to see that various people have different requirements and expectations. Personas aid in identifying the variety of people who will interact with the product because there are typically multiple user types. Personas assist designers in stepping outside of themselves and acknowledging that there are various people with various wants and expectations. Designers will be able to better predict what real people need by designing personas. Personas aid designers in concentrating on product development techniques. Knowing the user well makes it simple to ascertain for whom the product is developed and what features the user needs and does not need (Greer and Harris, 2018).

For this purpose, two personas are designed. One is of the user role one (pet owner) persona and the other is the role two user , i.e., vet persona. Pet owner in the current project is the person who owns the dogs. They are used to gather data related to dogs. This helps to check their health and thus records that data into the application. This further aids in understanding the basic needs of the pet owners as to how an application will be useful for them. Secondly the vet persona helps to identify the symptoms needed to analyse a disease.

**Pet owner**

Timeline

Description automatically generated

Figure 6: Pet owner Persona

**Vet:**

Text

Description automatically generated

Figure 7: Vet Persona

### 3.2.4 Use Case Diagram

After creating the user personas, a use case diagram was made to analyse the different user roles.

Diagram

Description automatically generated

Figure 8: Use Case Diagram

A use case diagram in UML serves to illustrate the various ways a user may interact with a system. A use case diagram in the Unified Modelling Language (UML) can simplify the characteristics of your system's users (sometimes referred to as actors) and their interactions with the system. Below figure represents the use case of the application built to check the chronic condition in dogs. As already described, there are two user roles for this application one is the pet owner and other is the vet. The functions that the application should provide for both types of users are described very well in the use case diagram like who can login, add pets, track symptoms, send reports or send feedback, etc. This diagram represents the user stories which lead to identification of the user requirements at the initial stage.

### 3.2.4 Functional and Non-Functional Requirements

As shown in the use case diagram created after interviews and market analysis, the following functional and non- functional requirements are gathered that a user needs from this application:

* Login/ Register into the application
* Different application structure for pet owner and vet
* Adding Pet, its picture and other details like age, breed, weight, walking habits, etc
* Adding multiple pets
* Tracking severity of disease by checking symptoms
* Consulting with vets
* Show health records of pets
* Compatible for both android and iOS platform
* Sending Notifications to users for submission of report and getting feedback
* Faster and Reliable
* Provide nearby list of vets
* Scalability and responsiveness
* Security and Compatibility

**Functional Requirements**

The following table shows the prioritization of functional requirements.

Table

Description automatically generated

Chart, waterfall chart

Description automatically generated

Figure 9: Functional requirements

**Non-Functional Requirements**

Below are the non-functional requirements that are gathered from interviews and analysis with explanation and status of result regarding the application**:**

**a)Performance:-**The performance of an application can be determined by its response time, i.e. how long it takes to complete a given task.

For Instance, when the application starts, it should not take more than 3 seconds for the splash screen to load. Developers should also ensure that your application does not interfere with user input.

Implemented : Loaded first screen after splash within 3 seconds.

**b)Scalability :-**Apps need to use more often or process more data over time.

For example, as user data grows (cache, stored data, etc.), the application must be able to process it without delay so that it can optimize how it is stored and accessed.

Implemented: checked with more than 1000 dummy records.

**c)Responsiveness: -** Applications must respond to the highest priority user input or external interrupt and return to the same state.

Example: - If the application is aborted by a call, the application should be able to store: Keep the state and return to the same state/page before the page was broken.

Implemented: Loaded same screen after any interruption.

**d)Usability: -** Users should be able to easily understand the flow of the application. Users should be able to use the application without the advice or assistance of an expert/guide. If you have to describe the user experience, it's bad UX.

Implemented: All participants confirmed in their second round they liked the application.

**e)Reliability: -**Application must be trusted for business purposes. This means that when a user performs an important action, it must be confirmed by confirmation.

Implemented: Implemented as per one user’s feedback.

**f)Security: -** All application data must be protected and encrypted for minimal protection from external and internal attacks.

Not Implemented : Future perspective

**g)Availability: -**There should be a common plane where users can access the application to install regular updates, search for and leave feedback.

Examples: - Apple App Store and Google Play Store.

Not Implemented : Future perspective

**h)Screen** **Responsiveness: -** Since many mobile devices today have different screen sizes and layouts, apps must be able to display their layouts for different screen sizes. With automatic adjustment of font size and image rendering.

Implemented: Support all screen sizes of Android and iOS devices.

The following table represent the prioritization of these requirements:

Table

Description automatically generated

Figure 10: Non-Functional requirements

# 4. Design

Design refers to the plan and specification for the construction of a system. From the architecture of the user interface to the code that transforms requirements into functionality, it consists of both high-level and low-level components. In addition to its functionality, design is also about how it looks and feels. In this process, problems are articulated and solved continuously. It differs from other systems in that it sees development through layers of ideas and requirements, architecture (high-level design), design (low-level design), and finally code. The process of abstraction is implemented by encapsulating code and data into classes and functions.

## 4.1 System Architecture

An architecture defines the structure, behaviour, and more views of a system. A system architecture description is a formal representation and description of a system organized to support reasoning about its structure and behaviour. When creating a quality mobile application architecture, you need to keep all application processes and functions in sync. It remains a big question which features future applications will really need. So including all ideas in the first version is not the best solution.

This increases development time, overloads the user interface, and delays the product launch "battle". Below figure shows a model for data systems in mobile application architectures which was followed during application development :

Diagram

Description automatically generated

Figure 11: System Architecture

### VIPER Architecture

VIPER Architecture is the design pattern used for designing this system. VIPER is a design pattern which is used to develop the various iOS applications. It is considered to be an alternative for the several design patterns such as MVC and MVVM. It offers a comparatively good abstraction layer which results in a relatively more scalable code. VIPER can go an extra mile for the purpose of achieving the responsibility principle which states that each and every component should be responsible for each and every functionality. VIPER includes multiple functionality such as the View, Interactor, Presenter, Entity, and Router. Further VIPER is considered to be the long term solution in the development. Therefore it is used in the current project as well. Moreover, it aids in giving a tighter architecture which is considered to be more reliable, thus resulting in less error.

Diagram

Description automatically generated

Figure 12: Viper Architecture

## 4.2 User Interface Design

User interface (UI) design is traditionally associated with what we think of as design. User interfaces are how systems are accessed by users. Some primary methods were used for the design process. User Personas and requirements were used to visualize who is using this application and what functionality they require from this application. Through sketching and ideation, potential solutions are generated, while storyboarding and mapping guide the system's navigation. Comparative research analyses best practices and the advantages and disadvantages of similar systems existing in the market. The final step is prototyping, which not only creates a solid design plan but also allows for testing and feedback.

First of all users have to identify what the application can do and how to use it properly. In the book “*The Design of Everyday Things”,* the author *“*Don Norman “ tells 7 principles for good design. These are visibility, consistency, affordance, mapping, feedback and constraints. (Norman, 2013). *Dog Care* design aimed to implement these properties.

The user experience design process involves analysing users and their requirements, sketching and storyboarding, building prototypes, and then implementing UX research methods. The first step is identification of users and their needs from the system. After identifying the user requirements, some sketches of the design were designed to understand the potential functionality and navigation of the application. This became the application’s low-fidelity prototype culminating with the first sprint.

### 4.2.1 Low-Fidelity Sketches

Whenever it comes to UX design, designs are crucial. A decent design emerges with a slew of extra benefits for the application. From idea verification to consumer and user experience testing, prototypes can help accelerate application initiatives to make things easier in the future. While a prototype can assist users in verifying a concept, wanting to see what that idea will appear like in real-time, it also brings focus to lacking characteristics and mandates perseverance at a preliminary phase.

The following figures represent the sketches for the Dog Care application that were used to create wireframes which served as low fidelity diagrams. Figure 12 shows the basic functions and navigation of the application as per the user stories. The user needs to add a pet record and check the symptoms of disease in the pet using this application, so this sketch represents the same features.

**A picture containing text, indoor, whiteboard

Description automatically generated**

Figure 13: Low fidelity sketch of application

**Text, letter

Description automatically generated**

**Text, letter

Description automatically generated**

Figure 14: Low fidelity sketches of different screens of application

Figure 13 shows low fidelity sketches for the screens in applications like user profile screen, pet record screen, settings screen, about us and privacy policy screens. The low fidelity screens were easy to design and change as they were made on paper. After these were finalized, wireframes of these screens were created which are explained in the next subsection.

### 4.2.2 Mid Fidelity Wireframes – Dog Owners

Balsamiq application was used to convert the sketches into wireframes which lead to the mid fidelity diagrams. These wireframes present the low fidelity sketches into a more professional representation.

Text, letter

Description automatically generated Graphical user interface, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

Figure 15: Detail Screen, Symptom Tracker, Conclude Screen

Above figure shows wireframes of detail screen, symptom tracker and conclude screens. Detail screen shows individual dogs with captured images and other data like name age breed. The symptom tracker screen gives an idea about what a symptom tracker looks like, with dog name on top left corner and diseases category on right followed by symptoms and their severity marker. Similarly the next sketch shows a concluding screen where the concluding message, send report, export report and nearby vet functionality is available.

### 4.2.3 Mid Fidelity Wireframes – Vet role

Figure 16 shows wireframes for the vet role. The vet role application starts with multiple requests screen received from dog owners that includes brief details of the dog name, diseases category and a message. Followed by the detail screen having additional information related to symptoms, pictures and an option to send a reply to dog owner, next is the same detail screen with request status changed like pending to replied.

A picture containing text

Description automatically generated Text, letter

Description automatically generated Text, letter

Description automatically generated

Figure 16: Dashboard Vet App, Report Screen(Replied), Report Screen(Pending)

## 4.3 Data Design

Data design produced an abstraction-level model of data. A computer-based system uses this model to refine it into a more implementation-specific representation. Data structure is one of the most important aspects of application software design. For this research, a flowchart of the application was created for structuring the data.

### 4.3.1 Flowchart:

Diagram

Description automatically generated

Figure 17: Flowchart of application

The above figure shows the flowchart of the application. With the help of use case diagrams and the low fidelity diagrams, the entities were identified for this application and these entities were used to create a flowchart of application structure. For this application, the main data entities are the login/registration data, pet data, symptoms data, result data, report and the feedback data. Relationship between all data entities is shown in the flowchart. As seen in the figure, when the user opens the application, the main screen initialization phase starts, i.e., the login or registration process. During this phase, all the initial variables along with the back end and front end of the application are started and placed where required. Then the pet data is added by the user and stored at the backend.

Once the application starts, a front screen appears where several tabs are available which represent the different symptoms. The user has to click on the symptom or, in different cases, multiple symptoms. Once all the symptoms are selected, the user has to click on the submission button. After that, the system analyses the symptoms with the help of the different cases and, as a result, gives an option to the user for sending a report to the vet registered with the app or exporting the report.

In this application, there are two different user roles, one is the pet owner and other is the vet. The application behaves differently for both roles. The pet owner role has the option to add pets, track symptoms and send reports to the vet. On the other hand, the vet role has the option to analyse the report and send feedback to the pet owner.

## 4.4 High Level Design

High fidelity design focuses on embedding the wireframes into a prototype close to the final product. These are highly functional and interactive. In the later stages of a project, prototypes are used to test usability and identify workflow problems. These are built using the final technology, i.e. , the Flutter technology (explained in next subsection) and PHP with SQL server. Detailed descriptions of the technologies and UI components utilized to implement the system's requirements will follow in the implementation section.

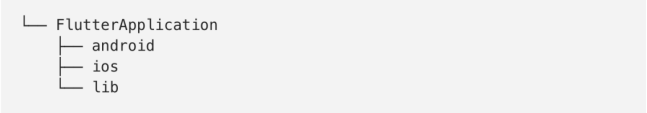
## 4.5 Technology and Languages

As per the study given by (Kuzmin, Ignatiev and Grafov, 2020), our daily lives are becoming more and more impacted by mobile applications. Since November 2016, mobile devices (48.19%) have generated more network traffic than desktops or laptops (47%). In order to be shared with a lot of users, a mobile application needs to introduce itself with two of the most important and independent platforms which includes Android and iOS. Due to a lot of differences between these two platforms, it is frequently necessary to build alternative skill sets. Numerous businesses have already considered and created cross-platform frameworks that resemble React Native and Flutter.

After doing extensive research on the arthritis in dogs and the existing applications on it, it was found out that there are very few of the applications which are available on this disease in dogs. Most of the applications built for the mobile platforms are not the actual tools which are used for arthritis but rather just a simple thing or the information to explain the arthritis. A simple and an extremely easy to utilize for an application is to help the owners of the dogs and provide them the way which aids in having the visual aspect of the arthritis disease. The application could help in providing a step towards the formulation of better visual solutions for arthritis. The application primarily targets the tendencies of the arthritis in dogs and the symptoms which helps to measure the severity of the arthritis in dogs. Frameworks which are available in the cross platform that are like the react native and flutter are discussed. For the current project, the flutter framework is used with the DART language.

### 4.4.1 Flutter

A cross-platform framework called Flutter focuses on creating high-performance mobile applications. Google publicly launched Flutter in 2016. Flutter applications can also be used on Fuschia in addition to operating on Android and iOS. Google has decided to use Flutter as the application-level framework for its upcoming operating system. Flutter is unique as it relies on the OEM widgets on the device rather than consuming web views. Each view component is rendered by Flutter utilising a powerful rendering engine. Building applications that are as high-performance as native programmes is now possible by flutter. Below figure helps to demonstrate the application structure of Flutter (Wu, 2018).



Uncompromised speed is achieved by writing a flutter project in the Dart programming language and AOT (Ahead-of-time) compiling it to the native platform architecture. To create the most typical interface elements that are utilized on the iOS and Android platforms, flutter offers widgets at the top level that are made up of numerous widgets. Developers can create their own widgets by combining other widgets at any level of the layered architecture because flutter adheres to an open and transparent architecture. This is due to its numerous advantages, which include quick app creation and concurrent app updates for iOS and Android. It is a good opportunity to determine whether Flutter is still a potent programming language for creating mobile applications in 2022 after many updates to it over the years. There are many benefits of the flutter which includes that you can release app updates simultaneously because the codebase is the same for both platforms. When there are separate projects for each platform, it is difficult to accomplish this. Synchronizing is challenging due to developers' inconsistent availability or prolonged platform implementation. When changes to the backend infrastructure are necessary as a result of an application upgrade, the issue gets worse. This issue is eliminated by Flutter and other cross-platform solutions because it is used for creating applications for both platforms in a simultaneous manner. One codebase may be used to run Flutter on both iOS and Android because it is a cross-platform solution. Naturally, it won't be reduced in half, but it's a good bet that writing a single codebase for both systems will require at least a 50% reduction in development time compared to writing it separately for each platform (Jadimath et al. 2022). For Flutter online applications in Android and iOS mobile apps, the Backend less SDK is provided.

### 4.4.2 DART

Every application in Flutter is created using Dart. A programming language named Dart has been created and is maintained by Google. It is widely used at Google, and it has proven to be capable of producing large-scale web applications like AdWords. Dart was initially created to succeed and replace JavaScript. Even though only a few other systems use reactive views, the Flutter application updates the view tree with each new frame. This behaviour has the downside of producing a large number of objects, some of which may only last for a single frame. Dart is a contemporary programming language that has been enhanced to handle the current scenario of the application which is used to check the health of the dogs and the severity level (Boukhary and Colmenares, 2019).

Dart, a language designed for quick apps on any platform, powers Flutter. Dart resembles Java, Kotlin, Swift, and TypeScript in appearance. Dart is tightly typed, but because it can infer types, type annotations are not required. The optimal approach is to utilise strong typing for functions and public variables, and var and type inference for local variables. Dart lacks the keywords public, protected, and private in contrast to Java. An identifier is private to its library if it begins with an underscore ( ). On any platform, Dart is a client-optimized language for creating quick apps. With a configurable execution runtime platform for app frameworks, it aims to provide the most productive programming language for cross-platform development (Tashildar et al. 2020).

For mobile, desktop, and back-end applications, Dart can compile to ARM and x64 machine code, as well as to JavaScript for the web. Dart's debug mode enables you to instantly reload active apps after making changes to the source code. JIT (just-in-time) and AOT (ahead-of-time) compilers are both available for Dart.

### 4.4.3 PHP (Backend)

The PHP project began as a small open source project that evolved as more and more people discovered its usefulness. A recursive acronym for PHP: Hypertext Preprocessor, PHP is a server-side scripting language embedded in HTML developed by Rasmus Lerdorf in 1994. PHP is used to handle databases, dynamic content, session tracking and to build entire websites.It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

I have used PHP with MySQL for creating the web services in the application and for managing databases.

A PHP module that is compiled as an Apache module on Unix is refreshingly fast in its execution. Despite the complexity of MySQL queries, even very large result sets can be processed in a record time once MySQL is started. Many major protocols are supported by PHP, including POP3, IMAP, and LDAP. First n-tier development is now possible with PHP4's support for Java and distributed object architectures (COM and CORBA).

A computer system must be installed with three essential components in order to run PHP Web pages. These are:

Web Server − Most commonly used is Apache web server, which is free and works with virtually all Web Server software, including IIS from Microsoft.

Database − The PHP programming language is compatible with virtually all database software, such as Oracle and Sybase, however the most commonly used database is MySQL that is freely available. The same is used in my application as it is freely available.

PHP Parser − An HTML parser is required in order to convert PHP scripts into HTML output suitable for web browsers.

# 5. Implementation

## 5.1. Build

As discussed in the previous section, Flutter technology with dart language was chosen for implementing this system. Android Studio was used for developing applications in flutter that can run on both android and iOS platforms. For backend, PHP was used with SQL server as database. The database schema is described below.

**Backend of the application (Database schema)**

Database schema design determines how data is organized into discrete entities, how relationships between organized entities are created, and how constraints are applied to data. Designers create database schemas so that data can be logically understood by other database users, such as programmers and analysts.

As per the flowchart, the figure 18 represents different entities like users, pets, symptoms, vets, requests and their connection with the primary key.

The database tables are shown in the appendices of this report which shows how the data of the application was saved and for each entity, a different table was created to manage the data properly. As already discussed in this report, My SQL server was used to create these tables and entities.

For saving data in these tables, APIs were created for the application using PHP language which are also mentioned in the appendices.

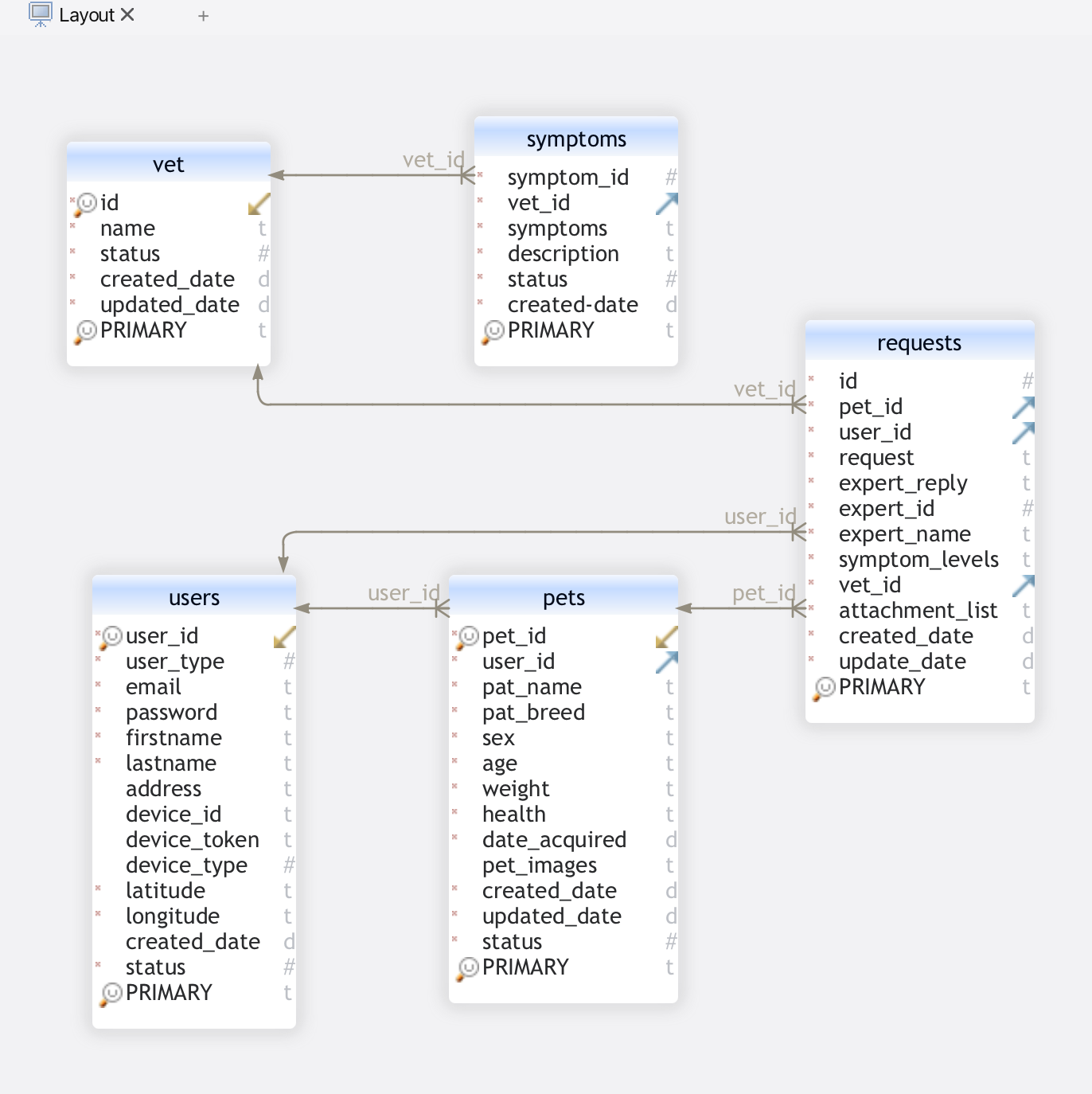


Figure 18: Database Schema

**Front-end of application (Screens)**

Using Flutter technology, the following screens were created for this application.

### 5.1.1 Login/Registration screen

Graphical user interface, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

Figure 19: Login / Registration

The above figure represents the login screen of the application. This is the initial screen. Therefore, there are multiple functions available on this screen. If the user of this application is new, then he/she must use the "Create Account" function. In this function, the user has to enter the essential details in order to register for this application.Once the user has successfully created his/her account, then the user can login into this application with the help of the email ID and password. This step is important because in this application, there is the data of the different users or some essential data such as address, phone number, and many more, which is important for both the user and the application. Since we need to keep data safe for multiple entries as per user’s acceptance during information gathering interviews, that’s why there is a login system in this application. If the user forgets his/her password, then, with the help of the "**forgot password**" option, the user can either retrieve the password or change the password according to its requirements. In such instances, the user gets the password on their provided email.

### 5.1.2 Reset Password

If the user forgets their password, they can visit the login page and if they are unable to log in with their username and password, they may be required to reset their password.

Graphical user interface, text, application

Description automatically generated

Figure 20: Reset Password

By clicking on the “forget password” textbox on the login screen, a user will be taken directly to where they can reset their password. If a user doesn’t know their password and needs assistance resetting it, click on the "Forgot Password? Email Address" link under "Reset Password."

### 5.1.3 View/Update Profile

The user can update their profile information or change their name, email, etc. It's easy to change your contact information in the Profile section of the account.

Graphical user interface, text, application, email

Description automatically generated

Figure 21: View/Update Profile

To update the profile information, click Settings and then click User Info. The Update Profile button will be highlighted in blue. If they want to change your password, click Settings and then click User Info. The Change Password box is on the right side of the screen under Security Questions. Enter the current password, type a new one twice (to make sure it's correct), then click Save Changes.

### 5.1.4 Home Screen

The below figure represents the main screen of the application.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 22: Home Screen

Once the user successfully logs in to the application, this initial screen can be observed by the user. This screen shows the user name along with the previously added dog data or registered dog data (the registration screen process is discussed later in this section). A user can also add the new data or create a new pet with the help of the (+) button in the middle of the screen. A user can directly shift to the request option by tapping on the person's logo, available in the middle at the bottom. Similarly, a user can also move to the setting options directly from this screen by tapping on the setting options logo.

### 5.1.5  Create Pet

Since this application is for the dogs, therefore, their data is important in order to track their health or disease symptoms. Once the user completes the "Login" procedure, he/she can add a number of pets to track their health.

Graphical user interface, text, application

Description automatically generated

Figure 23: Create Pet

The above screen shows the data form for the pet. This form is for a single dog, which means for every dog there is a different form because data varies from person to person. In this form, the user has to enter different data like pet name, breed, and sex, date of birth, weight, and health. The user can also add a photo of the dog in order to differentiate between the multiple same-breed dogs. Once the above procedures are completed, like login and form filling, the general data of the dog is saved in this application for future use.

### 5.1.6 Dog Details

Below figure shows details related to dog for example dogs picture taken at the time of creation dog profile, date of birth, breed, sex, weight and health. Also there is an option to go on the symptom tracker screen.



Figure 24: Dog Details

### 5.1.7 Symptom Tracker Screen

The below screen represents the Symptom Tracker Screen. This screen is the most important screen in this application because all this application's work is based on this screen. As it can be observed, different numbers (0–3) are assigned to define the severity of the symptom, where 0 means never and 3 means very often (severe).There are a number of predefined symptoms in this application so that a user doesn’t need to search for the symptom. The user has to increase the number by pressing the (+) button in order to define the severity. This application is based on a heuristic approach. Therefore, if a person accidentally increases the severity, the user can reduce the severity by pressing the (-) button. All these predefined symptoms are gathered from different literature reviews. If some symptoms are missed from this list, a user can seek help from the experts by using the request option.

Graphical user interface

Description automatically generated

Figure 25: Symptom Tracker

### 5.1.8 Conclusion Screen

The below screen depicts the applications or complete process's conclusion screen. As previously stated, this screen allows the user to determine whether or not the dog has arthritis. Similarly, a useful option on this screen is to send the report to the concerned person in order to gain a better understanding of the dog's health; or a user can export the dog's report in order to share that report out of the app network. A user can also see a number of experts in its vicinity, as well as multiple options such as whether it is open or closed, the distance between the user and the Centre, and the direction.

Graphical user interface, text

Description automatically generated

Figure 26: Conclusion Screen

### 5.1.8 Report

In the figure below the report of the dog is shown with two options: print and save options. Via print option application will access configured wireless printers with smart device and on click of save option report could be exported to phone memory or to other applications.

Graphical user interface, text

Description automatically generated with medium confidence

Figure 27: Report

### 5.1.9 Request Screen

If the symptoms of the dog are not mentioned, then a user can use this screen to contact a concerned person regarding the disease. Similarly, this section is also important because a user can directly contact the experts regarding their dog's health, the food they are consuming, sleeping pattern, and many other issues. The major aspect which differentiates this application with the rest of the application is that a user can directly share the current pictures of the dog with an expert so that the expert can view the current condition of the dog and determine if it needs treatment or medicine on an immediate basis. Similarly, they can prescribe or give the users instructions to prevent the diseases from spreading until they check them up physically.

Graphical user interface, text, chat or text message

Description automatically generated

Figure 28: Request

### 5.1.10 Settings screen – Pet Owner

Figure 29 represents the different setting options provided by this application. In this application, with the help of the user info, a user can get its general information which was added when the user created his/her account. Similarly, in the dogs' past health record tab, a user can visualize their pets' previous health with the help of the data.

Graphical user interface, text, application

Description automatically generated

Figure 29: Settings

Furthermore, this application is develop in order to help the dog owners and there are a lot of dog owners which are unaware about the diseases related to the dogs, therefore, in this application an easy method of sharing app is available by using this, a user can share this application with its colleagues, friends, and family. The last option on this setting page is the most important because, under the experts list tab, a user can observe a number of experts. They can be a vet, professional, or experienced person. These people can help the user with their experience and, in some cases, they can provide their assistance physically, which differentiates this application from the other applications.

### 5.1.11 Symptom Tracker History

In figure 30 there are several symptoms that are often observed in the history of a dog.

Graphical user interface, text, application

Description automatically generated

Figure 30: Symptom Tracker History

### 5.1.12 All Request Screen – Vet Client

Below is a landing page of the vet – client app after the same login process as performed in user/dog owner app. Here the vet can see all requests received and share valuable feedback after evaluating symptoms and images of dog uploaded by dog owner while creating request from user client application.

Graphical user interface

Description automatically generated

Figure 31: All Requests - Vet Client

### 5.1.13 Settings Screen – Vet Client

Setting screen is also similar to user client app but with some limited functionalities for example it has only user info with update functionality, About app, Logout and share app options.

Graphical user interface, text, application, email

Description automatically generated

Figure 32: Settings - Vet Client

### 5.1.14 User Journey – User / Dog owner Application

Graphical user interface, application, Teams

Description automatically generated

Figure 33: User Journey - Dog Owner App

The diagram above depicts the user's journey, or user flow. When a user first enters the application, the first screen that person notices is the login screen; if the person is new to this application, they first create an account. The user can create an account by providing relevant information. Similarly, if the user is an existing one, by entering credentials, the user can proceed to the main part, and if the user forgets their password, they can be recovered using their email address.

The above diagram depicts the user's journey, or user flow. The Login screen is the first screen a user notices when entering the application; if the user is new to this application, they must first create an account. By providing relevant information, the user can create an account. Similarly, if the user is an existing one, the user can proceed to the main part by entering credentials, and if the user forgets their password, they can be recovered using their email address.

If the user wants to track the symptoms of their dog, they can enter the severity of the symptoms in the symptom tracker tab, and after completing all of the requirements, they can receive a conclusion about their dog's health and generate a report based on these symptoms to share with experts or vets. The best feature of this application is that a user can use it to share the same report with multiple experts in order to consult about the dog's health.

This user journey diagram is a complete flow of the whole application. As there are two roles in the application, so for both the roles, the user journey diagram is different. Pet owner flow is mentioned above, now I will explain the vet flow using the user journey diagram of the vet application.

### 5.1.15 : User Journey – Vet Client App

*Graphical user interface, application, Teams

Description automatically generated*

Figure 34: User Journey - Vet Client

## 5.2 Life Cycle and project Management

The development of the Dog Care application was completed through multiple phases following the Agile strategy for development. The overall process was carried out in Sprints which are explained later in this section. Before that, it is needed to understand first how the stages of development of this application.

**Life Cycle**

Mobile application development goes through multiple stages from planning to deployment. (Benington, H.D. (1956)

As seen in the below  figure, first of all the aim of the application was identified like this application aims to track symptoms and vet consultation.

Diagram

Description automatically generated

Figure 35: Life Cycle of application

After that market analysis and interviews were done to identify user requirements like getting login, adding pets, tracking symptoms, getting vet consultation and other functionalities. According to these requirements, some sketches were created which further converted into low fidelity diagrams and high fidelity diagrams for getting the idea of designs. After designing, the next step was to implement the designs using Flutter technology with Dart language on Android studio. After implementation, a second round of interviews with users was conducted for testing and evaluation of the application. After testing, the application was refined with suggested feedback of users and hence a final application was developed.

**Project management**

Following agile development, the development of Dog Care application was completed in almost 5 sprints where each sprint was 3-4 weeks. Each week, I used to attend meetings with my supervisor to discuss the work done so far and what to be done in coming weeks. My advisor used to give feedback on my work and according to that I make changes. In the first phase I developed a use case diagram that represents all the user requirements and using wireframes and design , I built an application using Flutter and PHP that includes all the features that were mentioned in the user stories. So, this is how a use case diagram that represents the user stories was finally converted to a mobile application using Flutter technology.

## 5.3 Coding Standards

A coding standard makes sure that all programmers using a certain language adhere to the set standards. As a result, the code is consistent and is simple to understand. Good software development companies expect their programmers to adhere to coding standards, which are a well-defined and standardised style of coding. A coding standard gives the codes a uniform appearance. It also minimises complexity while enhancing readability, maintainability, and intricacy of the code. It facilitates code reuse and makes error detection simple. It encourages good programming practices and boosts programmers' productivity (Sahu and Tomar, 2017).

In the development of this application, I used Dart language with proper syntax and a clear code is written to be easily readable and understandable. For this, a few conventions were used for naming and ordering packages, classes, methods and variables which are in line with best practices. We followed best practices for naming conventions, ordering classes, packages, classes and variables in this application.

### 5.3.1 Naming Conventions

Naming practices makes sure programmers are aware of the proper naming conventions for variables and classes to ensure that they are accurate and consistent. The names should be taken such that they are self-explanatory.

For example:

I have taken the “petDetail” variable to store details of a pet and the variable name describes the meaning itself.

Similarly, I have created the following method (saveConcluderesultAndSymptoms) to save concluded results and symptoms and the name of this method itself explains the function.

void saveConcludeResultAndSymptoms()  
{  
 ProgressManager.*showAlertDialog*(buildContext, "Save Info...");  
 Map<String,dynamic> concludeResultDic = Map<String,dynamic>();  
 concludeResultDic[MyApp.*appEngine*.sqlDBManager.concludeTableColumnSymptomsData] = jsonEncode(widget.dogSymptomsArray);  
 concludeResultDic[MyApp.*appEngine*.sqlDBManager.concludeTableColumnDogId] = widget.petDetail['pet\_id'];  
 concludeResultDic[MyApp.*appEngine*.sqlDBManager.concludeTableColumnVet] = 'Arthritis';  
 concludeResultDic[MyApp.*appEngine*.sqlDBManager.concludeTableColumnConcludedResult] = 'Dog health is normal no need to worried.';  
 concludeResultDic[MyApp.*appEngine*.sqlDBManager.concludeTableColumnCreatedDate] = maasssHelperGetCurrentDateWithGivenFormat('yyyy-MM-dd');  
 MyApp.*appEngine*.sqlDBManager.insertOneRowInTable(concludeResultDic, MyApp.*appEngine*.sqlDBManager.tableConclude);  
 ProgressManager.*showAlertDialogWithAutoDismiss*(buildContext, 'Successfully Send.', 2, 1);  
}

### 5.3.2 Class Ordering

In order to differentiate between methods, variables, and properties, the order of each section is crucial. These should be used in an order with an empty line for separation.

For example:

The dart functions, packages and frameworks are imported with an order and an empty line is provided in between their import commands. The “dart:” imports should be written before other imports.

import 'dart:async';  
import 'dart:convert';  
  
import 'package:dogcare/frameworks/managers/progress\_manager.dart';  
import 'package:flutter/material.dart';  
  
import '../frameworks/engine/app\_engine.dart';  
import '../frameworks/maasss\_helper.dart';  
import '../main.dart';

### 5.3.3 Formatting Guidelines

A standard format improves readability and understanding of source code. So, errors in the code can be avoided using this.

It's easy to format code with Dart's advanced automated code formatter, dartfmt provided by Dart language. The official whitespace can be removed using this code formatter.

Methods shouldn’t have more than an average of 30 code lines, excluding line spaces and comments. So, I wrote all the methods accordingly.

### 5.3.4 Comments

Comments should be mentioned in the code for easy understanding of the code and for code optimization

For example: The following code shows comments starting with “//” that tells what the line of code is doing to make the code easy to understand.

class DogDetail extends StatefulWidget {  
 // declare the variable to store the pet details  
 var petDetail = Map<String,dynamic>();

## 5.4 Sprints

Sprint is the main part of the agile methodology. A Software is created in iterative "sprints," with the goal of completing the desired functionality in each sprint's functioning prototype. The development of the current project took place over five main sprints, each lasting two to three weeks. Each sprint results in the creation of a prototype, which is then developed, built, reviewed, and tested. These continue until the product is ready for deployment and all "must have" requirements have been satisfied.

* In the **first sprint** that was of duration 4 weeks, the goal of the project was identified and feasibility was tested for the application. Literature review was done in this sprint with market analysis of existing projects. Additionally, interviews preparation were made during this sprint like ethical application, placing advertisement and taking consent of users for interviews. This leads to some of the initial requirements.
* In the **second sprint** (3 weeks), user interviews were conducted in which basic requirements were identified and user stories were created, the product backlog was implemented, and a low-fidelity prototype was created by sketching ideas for the user interface design.
* The **third sprint** saw the first high-fidelity prototype employing the technology produced while the initial criteria were re-evaluated based on the interview results. The infrastructure of the system was put into place by this front-end prototype.
* The functionality of the prioritized requirements were implemented in the **fourth sprint**, which was built on the previous prototype.
* The prototype was reviewed and further functionality was added during the final and **fifth sprint**. During this sprint, a second round of user interviews and testing were also carried out. Based on the evaluation, the application was refined with suggested features.

# 6. Testing and evaluation

Testing is the procedure of examining and assessing the application's capabilities and performance and this is the final stage of the life cycle of the application. There are various types of testing that examine the app's backend in addition to the user interface and design of the user experience. When it comes to mobile, testing of functionality, usability, and consistency of an application should be done. Testing your app on mobile devices can be done manually or through automation. Testing affirms but also authenticates the development process. It detects bugs and assists developers, in refactoring, cleaning, and delivering a reliable system. Finally, testing is critical to providing a high-quality final product. As a result, two types of testing are carried out for this application in order to improve its user experience. These tests include usability and heuristic evaluation, which will be discussed further below.

## 6.1 Usability

Testing of the final proposed design, testing of usability was also performed in an attempt to assess the experience for users and the convenience of the capabilities and connectivity (or the cumulative understand-ability of the framework). Applicants were allowed to perform a variety of operations on the implementation for evaluation. Their movements were tracked and measured to identify any points at which they might become confused or make a discrepancy. The following tasks were assigned to participants:

1. Sign up for an account and log in to the system.
2. Add a new pet with pet details like name, age, breed, weight, sex, health status,etc
3. Add a pet picture.
4. Select the dog's symptoms from the symptom tracker.
5. Get the concluded report.
6. Save the report.
7. Send the report to the vet.
8. Give feedback to the report.

Various users perform the above operations based on their own needs, and the bugs discovered through these evaluations include limited character for name, spelling issues, lagging issues, and many more. All of these issues were resolved before moving on to the next round of testing.

## 6.2 Heuristic Evaluation

Once it is time to evaluate the structure of an application, Jakob Nielson (2019) suggested ten user interface heuristics that concentrate on how customers perceive, recognize, and utilize details. These are some examples:

The first heuristic is awareness of framework status. The application should always effectively keep consumers aware of what is going on by providing adequate comments in a timely manner. The match between the application and the actual world is the second heuristic. Instead of system-oriented aspects, the application should communicate the users' vocabulary, using utterances, idioms, and definitions that are familiar to the consumer. Adhere to real-world conventions by arranging data in a logical and natural sequence. User control, as well as independence, is the third heuristic. Consumers frequently select application functions by accident, necessitating the need for an explicitly labelled 'emergency exit' to exit the undesirable situation without having to engage in a lengthy dialogue. Erasing and reconfiguring are encouraged.

Continuity and benchmarks are the fourth and fifth heuristics. Consumers should not be required to guess whether various phrases, circumstances, or activities imply the same thing. Error deterrence is the fifth heuristic. A proper consideration that precludes an issue from arising in the first place is even more beneficial than acceptable error codes. Alleviate error-prone circumstances or verify them against them and provide consumers with a verification option before committing to an activity. The sixth heuristic is recognition instead of recall. Make components, activities, and choices visible to reduce the person's memory requirements. The consumer does not need to recall details from one section of the interaction to the next. When suitable, system instructions must be noticeable or readily available.

The seventh heuristic is usability and versatility. Unnoticed by the new consumer, accelerators may frequently expedite the engagement for the experienced consumer, allowing the application to accommodate both untrained and trained consumers. Allow customers to personalize frequently performed tasks. Expressive and simple design is the eighth heuristic. Dialogues shouldn't contain data that is inconsequential or only occasionally required. Every additional piece of detail in a conversation needs to be contested with the pertinent pieces of data, reducing their comparative perceivability.

The ninth heuristic is to assist users in recognizing, diagnosing, and recovering from inconsistencies. The error messages must be written in straightforward vocabulary (no abbreviations), describe the issue accurately, and provide insightful feedback to solve the issues. Assistance and supporting documents are the final heuristics. Despite the fact that it is preferable if the framework can be employed without recording, assistance, as well as documentation, may be required. Any such details must be simple to find, concentrate on the participant's task, including concrete measures to be taken, and not be excessively

At the completion of the finished proposed design, heuristic evaluations were performed by the specialists in the mobile application development field. Utilizing the previously mentioned heuristics, these specialists carried out a thorough read of the graphical interface, evaluating the capabilities. They were instructed to utilize the working model, examine its capabilities, and identify design flaws, noting their intensity and which of Neilson's heuristics they violated.

Design approach, removal of extra content, minimalist look, theme, severity adding style, and many other issues were highlighted by these experts. All of the issues that were highlighted were addressed with the user in mind. Various previously highlighted issues (usability issues) were also resolved by resolving these issues. Finally, the application's working capabilities and appearance improve.

## 6.3 Mobile testing

Mobile testing is important for a number of reasons. Mobile apps are essential to running your business. And it is very important to test your mobile application to make sure it works correctly.

**Types of mobile testing**

There are mainly three types of mobile testing

**a)Functionality**

Functionality of app includes

* Business Flow
* UI Testing (Including landscape and portrait mode of device)
* Coverage on cross platform

**b)Real Environment Testing**

Application has been verified on basis of below environmental testing scenarios

* Network Conditions
* Interruption from calls or messages
* Background / Foreground
* Touch / Drag Gesture

**c)Non Functional Testing**

Application also verified on below non-functional testing scenarios

* Performance
* Availability
* Scalability
* Usability

## 6.4 Unit testing

As part of software development, unit testing involves individual and independent testing of the smallest testable components of an application, called modules. This testing methodology is performed by software developers. During the development of the app manual unit testing was performed on various functionalities such as Login / Logout / Registration / Pet Creation / Report Sharing and Symptom Tracker.

## 6.5 System testing

System testing, also known as system integration testing, is the process of evaluating how different application components interact with each other in a fully integrated system or in an application. During the development of DogCare application system testing was done with the web API’s.

## 6.6 User acceptance testing

Usability testing is a popular UX research methodology.

In a usability testing session, the researcher asks participants to complete a task, typically using one or more specific user interfaces. As the participant completes each task, the researcher observes the participant's behaviour and listens for feedback.

### 6.6.1 Application Deployment

An app link was created on Diawi (<https://www.diawi.com>) for installing the application and was shared with users via messages and emails. Users were asked to install the app on their mobile device either android or iOS before the second round of interviews.

### 6.6.2 Users feedback and product refinement

In the evaluation process, 4 participants took part from Scotland , Germany and Abu Dhabi and shared their feedback on the application developed.

During the feedback interviews, most of the participants appreciated the application concept and found it useful for them.

Participant A from Scotland compared it with an already existing app available on market and said that “This is a unique concept.” Apart from that, other participants mentioned scope of improvements in terms of UI/UX as well as functional features as well. For example :

Participant B from Germany mentioned that the “The app should have a location based vets list along with an inhouse vets list”. He gave a reference to Google app he explained like google can show nearby restaurants app should be capable of showing nearby vets if available. Based on this feedback, the expert list screen was changed as shown in the following figure.

Graphical user interface

Description automatically generated

Figure 36:Expert List Updated Design

Participant B indicated that “the app should have a profile update feature as well where he can update not just name but he has an option to update his mail id picture and other information that he provided at the time of registration”. According to his suggestion, the profile screen was modified as shown below:

Graphical user interface, text, application, email

Description automatically generated

Figure 37:User Profile Updated Design

Participant C indicated that “Users should have confirmation alerts before performing actual actions for example on logout users should be notified again if they really want to logout or not.”

This is implemented and new design was modified as shown in the following figure:

Graphical user interface, text

Description automatically generated

Figure 38:Settings Updated Design

Participant D from Scotland mentioned another functional improvement he pointed to “If the application can have group chat functionality in future versions of the app so that one user of the app can share their dog’s experience and health symptoms to other users”. To explain his view, he took an example of a Facebook app. This change will be implemented in future.

On the other hand the participant from Abu Dhabi suggested “Whenever any user gets any feedback from vet or any vet gets a new request to analyse dog report from users they should get notified through application.” This change will be implemented in future.

In terms of UI/UK , the Germany participant mentioned that “The icons should be relevant to functionality as well as the main icon of the application should be enlarged”. He also mentioned that “the colour of information text should be normal black instead of red as red colour represents error or something wrong.” According to his feedback, changes were implemented and these are shown in below figures.

Graphical user interface

Description automatically generated

Figure 39:Symptom Tracker Updated Design

# 7. Conclusion and Future Work

## 7.1 Conclusion

To conclude, this research has met its aim to develop an application to track the symptoms of chronic disease (Arthritis) in dogs and provide remote consulting by analysing symptoms along with raising awareness among the pet owners. The research was completed and an application was created based on these goals going through different stages like identification of requirements from market analysis and interview, creating user stories from the requirements generated, sketching the requirements into wireframes using low fidelity diagrams, creating final prototype of application with high fidelity design, implementing the design using Flutter technology(Dart Language) at front end and PHP(with SQL server) used at backend to design a API’s, evaluating the application by conducting interviews and then finally refining the product according to users feedback.

The whole process was implemented using agile methodology and was completed in sprints. Each sprint was used to complete the stages of the project one by one. Each stage leads to the development of the next stage like the sketches were designed from the requirements and the high fidelity prototype was created from the designs. The study consisted of interviewing users in two rounds: The first part was related to gathering requirements and second was getting feedback on application. All the requirements that were collected in the first round of interviews like login functionality, monitoring symptoms, adding pets, consulting with vets, are met in the implementation of the application. In the testing phase that was carried out using the second round of interview, some more requirements were gathered like adding nearby vets, editing profile, push notifications and some other UI features.

Some of these feedbacks are implemented in the final application and many improvements are still possible with the help of these suggestions. These features/functionalities are discussed in the future work.

## 7.2 Future Work

This research is focused on the development of an app to support the management of chronic conditions in dogs. This application has features to identify Arthritis in dogs with the help of some symptom tracking. As of now, this application is focussed on Arthritis only. But we can add more diseases in future in this app. We can provide some more functionalities in this application like adding real time chat, selecting vet and some other UI improvements. So, there is scope of improvement in terms of functionalities and features in this application according to the feedback given by the participants and market analysis. Some of these are listed below :

### 7.2.1 Multiple Diseases Functionality

As suggested by one of the participants in interviews, the application should support more chronic diseases like cancer, Diabetes, kidney diseases etc. One of the participants from Scotland who was having one pet, suggested this feature in his interview. Currently, as per my research, I don't have this functionality but it will be added in future to make the app more versatile.

### 7.2.2 Registration of vet from admin panel

As of now, the application is open to register for both roles as dog vet and a dog owner. But dog vet roles can be misused to get realistic data and fake advice. To mitigate those risks registration of vet should be on admin panel and verified with some more authorised documents as well. So, in future, the app will be focussed on this security feature.

### 7.2.3 Group Chat Functionality

One of the users suggested developing a group chat feature in an application like in social media apps so that all users can connect with each other (if they allow) to discuss the symptoms and issues related to their pets with each other. This will be kept for future implementation as this feature will require use of other technologies that are time taking.

### 7.2.4 Notifications feature

One more basic feature comes out during information gathering interviews that is the application should support push/remote notifications to notify users in different scenarios based on whether they are vet or normal dog owners. One situation could be that a vet application would get notification once any dog owner raised any request or shared any report with inhouse vet. On the other hand dog owners would be notified when any vet suggested any procedure or shared any feedback through the app. The application currently supports local notification, but these push notifications will be implemented in future.

### 7.2.5 Real time chat with vet

From market analysis there should be a real time chat option to connect with any vet to get advice if a vet is available which will be implemented in future.

### 7.2.6 Freedom to choose vet

According to one more realistic approach, there should be an option to check the vet profile and select the vet on the basis of their expertise before sharing the dog's symptom report to them and consulting.

### 7.2.7 User Interface Enhancements

While working on the development of a custom mobile application, we have learned what can be done with an app and how it can improve people’s lives. We learned what features a hybrid application can provide and how the application can be made more interactive using various user interface options. So, in future UI improvements will be considered so that the application will become more attractive and user-friendly.

### 7.2.7 Deployment of application on AppStore and Play Store

The application developed using this research work is not available on Appstore or play store. This application can be currently downloaded using the diawi link. But in the future, we can upload it on the Appstore for iOS platform users and on Play Store for android platform users.

Considering all these features for future implementation, it is decided to make the application more versatile and user-friendly after spending some more time in research and implementation of this work.

# References

* Abdalameer, E.M.T., 2016. Pet Health Care System. journal of kerbala university, 14(3).
* Abdelghany, A.S., Darwish, N.R. and Hefni, H.A., 2019. An agile methodology for ontology development. International Journal of Intelligent Engineering and Systems, 12(2), pp.170-181.
* Ana, F.A., Loreto, M.S., José, L.M.M., Pablo, S.M., Pilar, M.J.M. and Myriam, S.L.A., 2020. Mobile applications in oncology: a systematic review of health science databases. International journal of medical informatics, 133, p.104001.
* Anderson, K.L., O’Neill, D.G., Brodbelt, D.C., Church, D.B., Meeson, R.L., Sargan, D., Summers, J.F., Zulch, H. and Collins, L.M., 2018. Prevalence, duration and risk factors for appendicular osteoarthritis in a UK dog population under primary veterinary care. Scientific reports, 8(1), pp.1-12.
* Basics: Mobile App Architecture & How to Start Building One in 2021, July 2020 [online]

<https://www.intellectsoft.net/blog/mobile-app-architecture/>

* Beerbaum, D.O., 2021. Applying Agile Methodology to regulatory compliance projects in the financial industry: A case study research. Available at SSRN 3834205.
* Benington, H.D. (1956): Production of large computer programs. In Proceedings, ONR Symposium on Advanced Programming Methods for Digital Computers, June 1956, pp 15--27.
* Berihun, B., Atnafu, D.D. and Sitotaw, G., 2020. Willingness to use electronic medical record (EMR) system in healthcare facilities of Bahir Dar City, Northwest Ethiopia. BioMed Research International, 2020.
* Bhanote, V.S., Pandey, A.K., Agrawal, A. and Manohar, S., 2022. A New Approach to E-Health Application Development Using Blockchain. In Big Data Analytics and Artificial Intelligence in the Healthcare Industry (pp. 213-235). IGI Global.
* Bibbo, J., Rodriguez, K.E. and O’Haire, M.E., 2019. Impact of service dogs on family members’ psychosocial functioning. The American Journal of Occupational Therapy, 73(3), pp.7303205120p1-7303205120p11.
* Boukhary, S. and Colmenares, E., 2019, December. A clean approach to flutter development through the flutter clean architecture package. In 2019 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 1115-1120). IEEE.
* Cendrós, J.M., Salichs, M., Encina, G., Vela, J.M. and Homedes, J.M., 2022. Pharmacology of enflicoxib, a new coxib drug: Efficacy and dose determination by clinical and pharmacokinetic‐guided approach for the treatment of osteoarthritis in dogs based on an acute arthritis induction model. Veterinary Medicine and Science, 8(1), pp.31-45.
* Chen, Y., 2021, August. Research on Android Architecture and Application Development. In Journal of Physics: Conference Series (Vol. 1992, No. 2, p. 022168). IOP Publishing.
* Christian, H., Bauman, A., Epping, J.N., Levine, G.N., McCormack, G., Rhodes, R.E., Richards, E., Rock, M. and Westgarth, C., 2018. Encouraging dog walking for health promotion and disease prevention. American journal of lifestyle medicine, 12(3), pp.233-243.
* Ebone, A., Tan, Y. and Jia, X., 2018, May. A performance evaluation of cross-platform mobile application development approaches. In 2018 IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft) (pp. 92-93). IEEE.
* FirstVet AB,2021. FirstVet [Online]

Available at:

<https://apps.apple.com/gb/app/firstvet/id1349241101>

* Greer, M. and Harris, H.S., 2018. User-centered design as a foundation for effective online writing instruction. Computers and Composition, 49, pp.14-24.
* ‌Harvard Business Review. (2018). Why Apps for Managing Chronic Disease Haven’t Been Widely Used, and How to Fix It. [online] Available at: https://hbr.org/2018/04/why-apps-for-managing-chronic-disease-havent-been-widely-used-and-how-to-fix-it.
* Hayat, F., Rehman, A.U., Arif, K.S., Wahab, K. and Abbas, M., 2019, July. The influence of agile methodology (Scrum) on software project management. In 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) (pp. 145-149). IEEE.
* Jadimath, M.V., Elemmi, M.C. and Dhumale, P., 2022. Mobile Application Development using Flutter SDK: Development of a News Mobile Application using Flutter and NEWS API. Journal of Computer Science Engineering and Software Testing (e-ISSN: 2581-6969), 8(1), pp.50-64.
* Janthajirakowit, C., Fongmanee, S. and Chaikhamwang, S., 2021, March. The Development of Application for Advice and Diagnosis of Diseases in Dogs. In 2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering (pp. 132-135). IEEE.
* Kamishina, H., Sugawara, T., Nakata, K., Nishida, H., Yada, N., Fujioka, T., Nagata, Y., Doi, A., Konno, N., Uchida, F. and Maeda, S., 2019. Clinical application of 3D printing technology to the surgical treatment of atlantoaxial subluxation in small breed dogs. Plos one, 14(5), p.e0216445.
* Kani, Y., Cecere, T.E., Lahmers, K., LeRoith, T., Zimmerman, K.L., Isom, S., Hsu, F.C., Debinksi, W., Robertson, J.L. and Rossmeisl, J.H., 2019. Diagnostic accuracy of stereotactic brain biopsy for intracranial neoplasia in dogs: comparison of biopsy, surgical resection, and necropsy specimens. Journal of veterinary internal medicine, 33(3), pp.1384-1391.
* Kim, G.M., Lim, J.Y., Kim, E.J. and Park, S.M., 2019. Resilience of patients with chronic diseases: A systematic review. Health & social care in the community, 27(4), pp.797-807.
* Kuzmin, N., Ignatiev, K. and Grafov, D., 2020. Experience of developing a mobile application using flutter. In Information Science and Applications (pp. 571-575). Springer, Singapore.
* Lankani, P. and Liyanage, W.L.S.V., 2021. Implementation of Smart Pet Care Applications in an IoT Based Environment. SPAST Abstracts, 1(01).
* Liyanage, W.L.S.V. and Wedasinghe, N., 2021. Implementation of Smart Pet Care Applications in an IoT Based Environment.
* Márquez Espinoza, E.G., 2018. Evaluación de cuatro variables clínicas de pacientes diagnosticados con parvovirus para prognosis de la enfermedad en la ciudad de Quito cinco años atrás (Bachelor's thesis, Quito: Universidad de las Américas, 2018).
* My Dog File and First Aid,2011. Dog Buddy Pro [Online]

Available at:

<https://apps.apple.com/us/app/dog-buddy-pro/id444082194>

* Nielsen, J., 2019. *10 Heuristics for User Interface Design.* [Online] Available at: https://www.nngroup./com/articles/ten-usability-heuristics/
* Norman, D., 2013. *The Design of Everyday Things.* New York: Basic Books.
* Pasha, M. and Shah, S.M.W., 2018. Framework for E-Health systems in IoT-based environments. Wireless Communications and Mobile Computing, 2018.
* Pasmanter, N. and Petersen‐Jones, S.M., 2020. A review of electroretinography waveforms and models and their application in the dog. Veterinary ophthalmology, 23(3), pp.418-435.
* ‌Requirement Gathering Methods Mamoun Eid November, 2015.
* Robinson, L., Lundgren, K. and Segal, R. (2019). Mood-Boosting Power of Dogs. [online] HelpGuide.org. Available at: https://www.helpguide.org/articles/mental-health/mood-boosting-power-of-dogs.htm.
* Romero, C.A.T., Ortiz, J.H., Khalaf, O.I. and Ortega, W.M., 2021. Software architecture for planning educational scenarios by applying an agile methodology. International Journal of Emerging Technologies in Learning (Online), 16(8), p.132.]
* Rudinsky, A.J., Rowe, J.C. and Parker, V.J., 2018. Nutritional management of chronic enteropathies in dogs and cats. Journal of the American Veterinary Medical Association, 253(5), pp.570-578.
* Sahu, D.R. and Tomar, D.S., 2017. Analysis of web application code vulnerabilities using secure coding standards. Arabian Journal for Science and Engineering, 42(2), pp.885-895.
* Schlag, A.N., Hayes, G.M., Taylor, A.Q., Kerwin, S.C., Dugat, D.R., Vitt, M.A., Stratton, D.T. and Duffy, D.J., 2019. Analysis of outcomes following treatment of craniodorsal hip luxation with closed reduction and Ehmer sling application in dogs. Journal of the American Veterinary Medical Association, 254(12), pp.1436-1440.
* Sheusi, J.C., 2020. Android Application development for Java programmers.
* Tashildar, A., Shah, N., Gala, R., Giri, T. and Chavhan, P., 2020. Application development using flutter. International Research Journal of Modernization in Engineering Technology and Science, 2(8), pp.1262-1266.
* Udell, M.A.R. and Wynne, C.D.L. (2008). A Review of Domestic Dogs’ (Canis Familiaris) Human-Like Behaviors: Or Why Behavior Analysts Should Stop Worrying and Love Their Dogs. Journal of the Experimental Analysis of Behavior, 89(2), pp.247–261. doi:10.1901/jeab.2008.89-247.
* Van Bourg, J., Patterson, J.E. and Wynne, C.D., 2020. Pet dogs (Canis lupus familiaris) release their trapped and distressed owners: Individual variation and evidence of emotional contagion. PLoS One, 15(4), p.e0231742.
* Vishwas Ng, 2019, Non-Functional Requirement of the Mobile Development system. [online]

[https://medium.com/@vishwasng/non-functional-requirement-of-the-mobile- development-system-e0ed98f2a872](https://medium.com/@vishwasng/non-functional-requirement-of-the-mobile-%20%20%20%20%20%20%20%20%20development-system-e0ed98f2a872)

* ‌Veterinary Practice. (n.d.). Effectively managing chronic conditions in pets. [online] Available at: https://www.veterinary-practice.com/article/managing-chronic-conditions [Accessed 16 Jun. 2022].
* Vizer, L.M., Eschler, J., Koo, B.M., Ralston, J., Pratt, W. and Munson, S., 2019. “It’s Not Just Technology, It’s People”: Constructing a Conceptual Model of Shared Health Informatics for Tracking in Chronic Illness Management. Journal of medical Internet research, 21(4), p.e10830.
* Wong, P.T. and Psych, C., 2016. How to write a research proposal. International Network on Personal Meaning, http://www. meaning. ca/archives/archive/art\_how\_to\_write\_P\_Wong. Htm.
* Woods, H.J., Li, M.F., Patel, U.A., Lascelles, B.D.X., Samson, D.R. and Gruen, M.E., 2020. A functional linear modeling approach to sleep–wake cycles in dogs. Scientific reports, 10(1), pp.1-8.
* Wu, W., 2018. React Native vs Flutter, Cross-platforms mobile application frameworks.
* Yuan, W., Nguyen, H.H., Jiang, L., Chen, Y., Zhao, J. and Yu, H., 2019. API recommendation for event-driven Android application development. Information and Software Technology, 107, pp.30-47.
* Zhang, Y. and Wang, J., 2018. Study on the Effect of Heat Clearing Chinese Medicine on Blood Physiological and Biochemical Indexes of Dogs with Canine Distemper.
* Zoetis Inc,2017. Pet Dialog [Online]

Available at:

<https://apps.apple.com/us/app/petdialog/id716717991>

* 11Pets Ltd,2021. 11pets: Pet Care [Online]

Available at:

<https://apps.apple.com/us/app/11pets-pet-care/id1232470530>

# Appendices

The appendices for this research work are attached separately in a zip folder named as “Supported\_Documents.” Following is the list of appendices that are provided in the zip folder.

1. Appendix A : Class Diagrams
2. Appendix B: API Collections
3. Appendix C: Database Tables
4. Appendix D: Interview Consent Forms
5. Appendix E: Interview Transcripts and recording
6. Appendix F: User Testing Form
7. Appendix G: Installation Guide