**Senior Project**



Primary Advisor: **Dr. Muhammad Haroon Shakeel**

Secondary Advisor: **Dr. Maria Tamoor**

Presented by:

Student Reg# Student Name

**22-10588 Ahmad Ibtehaj**

**22-10208 Hamza Khan**

**22-10658 Rabia Tahir**

Department of Computer Science

**Forman Christian College (A Chartered University)**

Topic of Project

**Senior PROJECT [LOST PET FINDER THROUGH VIEWPOINT INDEPENDENT IMAGE MATCHING]**

By

Ahmad Ibtehaj

Hamza Khan

Rabia Tahir

Project submitted to

Department of Computer Science,

Forman Christian College (A Chartered University),

Lahore, Pakistan.

in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE

IN

COMPUTER SCIENCE (Honors)

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Primary Project Advisor |  | Secondary Project Advisor |
|  |
| Senior Project Management Committee Representative |

# Abstract

*Finding a lost pet by circulating brochures is a time-intensive and futile task, which rarely results in much success. Our project aimed to design an online system to find lost pets with the help of artificial intelligence using digital image processing. The picture of the animal will be posted onto the platform by the owner of the pet and also by the person who found it, then our system will be able to find the highest similarity match and notify both the owner and the finder and share the details with them. A reliable automatic image matching is an integral component of such system. However, image matching is sensitive to the viewpoint of the taken photograph. We also aimed to develop a reliable approach for image matching that is viewpoint independent. Thus, we intended to make two contributions. (1) A platform for lost and found pets, and (2) development of a robust viewpoint independent image matching algorithm. We trained our AI models on large amount of data in order to get accurate results. This platform can be used by any type of user and has a very user-friendly GUI to keep things simple and easy.*

# Acknowledgement

We would like to acknowledge our advisor Dr Haroon Shakeel for motivating us to work on this problem statement and providing us with guidance and help whenever we were stuck in a problem during the duration of our final year project. It was his idea that we used for the final year project and he helped us in the training of our model and in the development of the website too. We would also like to acknowledge Usman Akram (a youtube teacher) whose tutorials made us learn react. He has a complete course available.

# List of Figures

Figure 1 Use case Diagram 20

Figure 2 Entity relation Diagram 23

Figure 3 Class Diagram 24

Figure 4 Component Diagram 25

Figure 5 DFD Diagram 26

Figure 6 Activity Diagram 27

Figure 7 Sequence Diagram 1 28

Figure 8 Sequence Diagram 2 29

Figure 9 Sequence Diagram 3 29

Figure 10 High Level System Architecture Diagram 31

Figure 11 Swimlane Diagram 33

# List of Tables

Table 1 UC-1 Sign Up 14

Table 2 UC-2 Login 15

Table 3 UC-3 Create A Post 16

Table 4 UC-4 Update Profile 17

Table 5 UC-5 Update Post 18

Table 6 UC-6 Delete Post 19

Table 7 TC-1 36

Table 8 TC-2 37

Table 9 TC-3 37

Table 10 TC-4 38

Table 11 TC-5 38

Table 12 TC-6 39

Table 13 Summary of All Test Results 39

Table of contents

[Abstract i](#_Toc61870443)

[Acknowledgement ii](#_Toc61870444)

[List of Figures iii](#_Toc61870445)

[List of Tables iv](#_Toc61870446)

[Chapter 1. Introduction 1](#_Toc61870447)

[1.1 Introduction 1](#_Toc61870448)

[1.2 Objectives 1](#_Toc61870449)

[1.3 Problem Statement 1](#_Toc61870450)

[1.4 Scope 1](#_Toc61870451)

[Chapter 2. Requirements Analysis 2](#_Toc61870452)

[2.1 Literature Review 2](#_Toc61870453)

[2.2 User Classes and Characteristics 2](#_Toc61870454)

[2.3 Design and Implementation Constraints 2](#_Toc61870455)

[2.4 Assumptions and Dependencies 2](#_Toc61870456)

[2.5 Functional Requirements 2](#_Toc61870457)

[2.6 Use Case Diagram 3](#_Toc61870458)

[2.7 Nonfunctional Requirements 4](#_Toc61870459)

[2.8 Other Requirements 4](#_Toc61870460)

[Chapter 3. System Design 5](#_Toc61870461)

[3.1 Application and Data Architecture 5](#_Toc61870462)

[3.2 Component Interactions and Collaborations 5](#_Toc61870463)

[3.3 System Architecture 5](#_Toc61870464)

[3.4 Architecture Evaluation 5](#_Toc61870465)

[3.5 Component-External Entities Interface 5](#_Toc61870466)

[3.6 Screenshots/Prototype 5](#_Toc61870467)

[3.7 Other Design Details 6](#_Toc61870468)

[Chapter 4. Test Specification and Results 7](#_Toc61870469)

[4.1 Test Case Specification 7](#_Toc61870470)

[4.2 Summary of Test Results 7](#_Toc61870471)

[Chapter 5. Conclusion and Future Work 8](#_Toc61870472)

[5.1 Project summary 8](#_Toc61870473)

[5.2 Problems faced and lessons learned 8](#_Toc61870474)

[5.3 Future work 8](#_Toc61870475)

[References 9](#_Toc61870476)

[Appendix A Glossary 10](#_Toc61870477)

[Appendix B Deployment/Installation Guide 10](#_Toc61870478)

[Appendix C User Manual 10](#_Toc61870479)

[Appendix D Student Information Sheet 10](#_Toc61870480)

[Appendix E Plagiarism Free Certificate 11](#_Toc61870481)

[Appendix F Plagiarism Report 11](#_Toc61870482)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Introduction

A user-friendly pet finder system with a reliable and authentic feedback system for the lost and found pets in the area. Presently, finding a lost pet by circulating brochures is a time-intensive and futile task, which rarely results in much success. Our project aims to tackle this problem with the help of Artificial Intelligence using digital image processing. The picture of the animal will be posted onto the portal by the owner of the pet and also by the person who found it, then our system will be able to find the exact match and notify both the owner and the person with their locations.

Most of us share a very loving bond with our animal companions and cannot dream of losing them ever. But pets straying away from their loving homes is a regular occurrence and a large majority never gets reunited with their owners. Among people who found a lost pet and tried to find their owner by contacting some animal shelter or through an advertisement in the local newspaper, only 38% of the finders succeeded in reuniting pets with their owners, with dogs much more likely to be returned to their owners (46%) than cats (3%) [1]. There also are many lost pet finding websites on the internet but they all have their own set of limitations and none of them incorporate automatic image matching in their search. Most platforms are only limited to uploading lost pet information or image and one has to manually go through all hundreds and thousands of lost pet records in hopes of finding what they’re searching for. The high inconvenience of the process hinders the objective of the site.

Every other household has a pet as a part of their family and them straying away and getting lost is a distraught occurrence every pet owner has usually faced. 1 in 3 pets gets lost at least once in their lifetime. To help our furry friends find their way back home, we aim to build an artificial intelligence (AI) based online platform. This platform would require both, the owner and the finder, to upload an image of the pet. The images could be from different angles and viewpoints for the convenience of its users. Through this mechanism, we can find the highest similarity index match between pet images uploaded on our platform. It would send a notification to both the owner and pet finder once a match is found. A reliable image matching algorithm, therefore, is the core of such a system. However, a dependable image matching in an angle and viewpoint-independent manner is challenging, especially in the case of pets. Thus, we also intend to work on devising a reliable matching algorithm.

Initially, we will be limiting the types of pets to dogs only, however, the system would be flexible enough to scale to the other types of pets as well. This platform would provide services worldwide.

## Objectives

Main objectives of this project are:

* Helping reunite lost pets with their rightful owners.
* Creating a dependable pet finding platform which uses image matching in an angle and viewpoint-independent manner.
* Devising a reliable matching algorithm to achieve maximum results.

The main objective of creating this platform is to build a user-friendly pet finder system with a reliable and authentic feedback system for the lost and found pets in the area. Presently, finding a lost pet by circulating brochures is a time-intensive and futile task, which rarely results in much success. Our project aims to tackle this problem with the help of Artificial Intelligence using digital image processing. The picture of the animal will be posted onto the portal by the owner of the pet and also by the person who found it, then our system will be able to find the exact match and notify both the owner and the person with their locations.

## Problem Statement

While there are several sites that offer lost pet finding services they have a lot of limitations, such as going through the posts and images manually, which prevents successful results. With the advent of artificial intelligence (AI) and computer vision (CV), it is possible to develop a pet lost and found system that is powered by an automatic image matching system in order to tackle all those limitations. However, a major hurdle in implementing such a system is the viewpoint (or angle) of the images that are to be matched and in the case of pets, this problem becomes even more challenging. For instance, an image taken from the front of a pet would most likely not match with the image of the same pet but taken from a different angle or viewpoint. Therefore, it is also crucial to develop a robust image matching algorithm that is not sensitive to the angle of the image while matching and can rely on minimum possible information. To this end, we intend to design a pet lost and found platform along with an AI-based reliable angle-independent image matching mechanism.

## Scope

Pets are the family we choose for ourselves and it’s extremely heartbreaking to lose them. But our furry friends are notorious for wandering off on their own. 70% of the US households have a pet, and although in Pakistan the stats of pet homes aren’t that high, they are steadily increasing with time. So, when the pets wander off, pet owners try to locate them by circulating brochures or through newspaper ads. A few websites have also been created to upload your missing pet details and then manually searching the existing results in hopes of locating your pet. All these methods have proved themselves not to be very useful. With our platform we aim to ease the process of reuniting lost pets with their owners. This platform would require both, the owner and the finder, to upload an image of the pet. The locations will also be tracked to improve result accuracy. The images could be from different angles and viewpoints for the convenience of its users. Through this mechanism, we can find the highest similarity index match between pet images uploaded on our platform. It would send a notification to both the owner and pet finder once a match is found. A reliable image matching algorithm, therefore, is the core of such a system. However, a dependable image matching in an angle and viewpoint-independent manner is challenging, especially in the case of pets. Thus, we also intend to work on devising a reliable matching algorithm.

# 

# Requirements Analysis

## Literature Review

Usually upon losing a pet, their owners go around asking in nearby streets or they pin lost pet brochures in their local vicinity or sometimes even put out advertisements in their local newspapers with rewards for finders. All these methods could be a hassle, expensive, and usually don’t have very high success rates. In the last couple of years, many lost pet platforms have been created to help find lost pets. Lost pet owners can just go and browse through all thousands of images present on the site of randomly found pets in hopes of finding their own pet among them. Manual searching process is long, exhausting and in most cases goes in vain. Lost pet information can be uploaded on those sites without any reference image too which reduces the chances of a successful finding. [2] Moreover, these platforms are region-specific and provide their services in certain areas only. It is a disadvantage for pet owners who are deprived of these resources due to the location barriers. Some platforms use facial recognition through image matching to find lost pets but the images need to be taken from the front for it to work [3]. With the angle agnostic image matching our platform offers a better chance at finding a perfect match. The already existing platforms don’t provide results based on the nearest possible locations. Our platform would provide an optimized search based on location services too and is free to be used in any part of the world without any region or monetary restrictions.

## User Classes and Characteristics

This product will be very helpful for all the pet owners and will ease their worries of ever losing their pet. This website would be user friendly so simply anyone can use this website for lost pet finding. Our user classes would be general public including pet owners and those people who find a lost pet. Other than this, our users won’t be limited to a certain region the website can be accessed by users anywhere in the world and image matching results will also be optimized based on nearby regions. Everyone on the website is favored. Our classes would be person and pet class. The Person class will have attribute User ID, password, login status, registered date. The method of Person class would verify login. The user and admin class will inherit the person class. The user class would have methods: register, login, update profile, add post, update post, delete post. The admin class would have methods: update database, ban user, delete user, write queries, and update user profile. The pet class would have attribute type, id, color, breed, age, image.

## Design and Implementation Constraints

* As everything is voluntary so there are no such legal implications
* One constraint is that we will be using google collab where we would only be allowed limited data usage per day.
* Another constraint is we need a python supported server on the backend. We cannot use any other.

## Assumptions and Dependencies

**Some of the assumptions regarding platform usage:**

* User has active internet connection.
* User should have an operating system with browser compatibility.
* User should have a decent camera.
* User will upload the image of the pet.
* User is computer literate.
* User will allow sharing his location (no privacy issues)

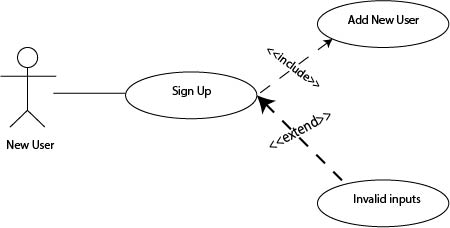
**Assumptions regarding viewpoint independent model development**:

* We have enough dataset to train our model (the data should have a large variety of angles and pets for best results)

## Functional Requirements

### Sign up

**Diagram:**

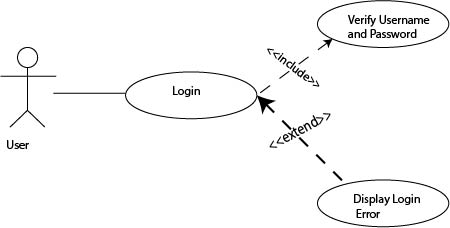


**Table 1: UC-1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-1 | |
| **Purpose** | | Allows user to create new profile | |
| **Priority** | | Medium | |
| **Pre-conditions** | | NIL | |
| **Post-conditions** | | User is on the landing page of website | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Visit the website | | Prompts user to log in OR sign up |
| **2** | User clicks on Sign up button. | | Sign up window is shown to the user along with input fields. |
| **3** | User enter his/her details. | | Store the info in the database and do authentication checks |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Provides wrong information (username/password) | | Authentication fails and user is asked to enter correct information |

### Login

**Diagram:**

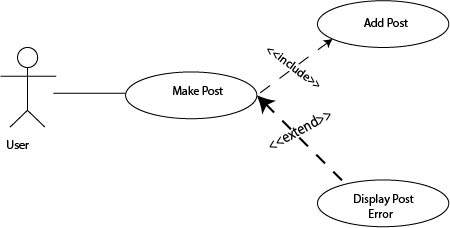


**Table 2: UC-2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-2 | |
| **Purpose** | | Prompts user to Log in | |
| **Priority** | | Medium | |
| **Pre-conditions** | | NIL | |
| **Post-conditions** | | Directed to home screen | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Visit the website | | Prompts user to log in |
| **2** | Enter login details | | Check the details and verify them from database and then loads home screen. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Provides wrong information (username/password) | | Authentication fails and user is asked to enter correct information |

### Add a post

**Diagram:**

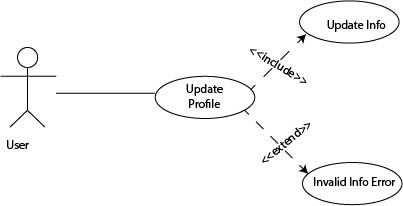


**Table 3: UC-3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-3 | |
| **Purpose** | | User to upload a post for search | |
| **Priority** | | Low | |
| **Pre-conditions** | | User is logged in | |
| **Post-conditions** | | User is on the home screen | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Clicks on Add a Post button | | Prompts user to select images from the device and fill information |
| **2** | Select the file from the device and enter details | | Upload the image onto the database and look for matches. User is notified of the search result |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Select wrong file type | | Checks for the file type and prompt user for invalid file. |
| **2** | Enters invalid information | | Checks for validity and prompts users to enter correct details. |

### Update Profile

**Diagram:**

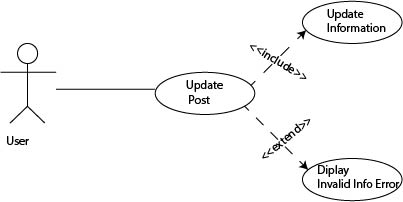


**Table 4: UC-4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-4 | |
| **Purpose** | | Update the information of the user | |
| **Priority** | | Low | |
| **Pre-conditions** | | User has an already existing profile | |
| **Post-conditions** | | User is on the home screen | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Clicks on Update profile button | | User is taken to a new screen with existing profile details |
| **2** | New information is entered | | Do authentication checks and update the info in the database |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Wrong info entered. | | Authentication fails and user is asked to enter correct information |

### Update Post

**Diagram:**

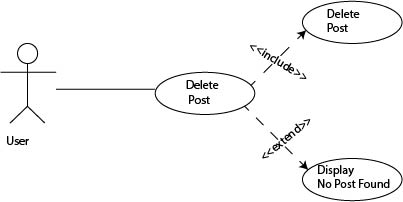


**Table 5: UC-5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-5 | |
| **Purpose** | | Update the information of the post | |
| **Priority** | | Low | |
| **Pre-conditions** | | User has an already existing post/posts | |
| **Post-conditions** | | User is on the home screen | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Clicks on Update post button | | User is taken to a new screen with existing post details |
| **2** | New information is entered | | Do authentication checks and update the info in the database |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Wrong info entered. | | Authentication fails and user is asked to enter correct information |

### Delete Post

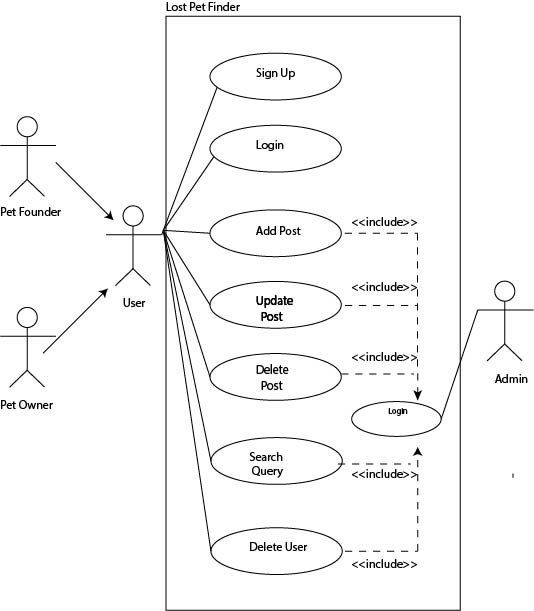
**Diagram:**



**Table 6: UC-6**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-6 | |
| **Purpose** | | Delete the post | |
| **Priority** | | Low | |
| **Pre-conditions** | | User has an already existing post/posts | |
| **Post-conditions** | | User is on the home screen | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Clicks on Posts button | | User is taken to a new screen with existing post details |
| **2** | Clicks on Delete Post | | Delete the post details from database and prompts user |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Clicks on Posts button | | System check and user is prompt with “no posts available” message |

## Use Case Diagram



**Figure 1: Use case Diagram**

## Nonfunctional Requirements

### Performance Requirements

As we have mentioned above that our website will be a reliable and flexible product so we need to make sure that everything that the user needs is always available to them for example the data and the services of the website. If we consider the authentication part, then after the user enters his/her credentials, he/she should be logged in after 1-2 seconds. Any delay more than 1-2 seconds will have a negative impact on our website. Since our website is an image matching based website, so the results that our website will provide must be accurate as there is no chance of mistake if we want our platform to grow. So, in order to avoid this thing, we need to train our model very efficiently and wisely and provide enough dataset to it. Also, the website shouldn’t take much longer while adding the details of the pet in the database and then take the user back to the landing page. We have also mentioned that our website will be user friendly so attractive graphical user interfaces will be provided to them so that the users don’t get bored or their attention starts diverting.

### Safety Requirements

Safety of this website is guaranteed for the clients by accepting the worst-case scenarios and afterward proposing appropriate measures for them. For instance, our item will be kept clean from any kind of explicit content or clicks on ads that lead to the downloading of malwares to give a protected and friendly impression while visiting the website.

### Security Requirements

Security is a very important component of any application and keeping it protected from different attacks must be on our priority list. Following protocols will be strictly implemented to secure our website:

* It is crucial to keep all platforms or scripts installed up-to-date.
* It is important to use strong passwords. To protect against brute force or SQL injections, passwords should be complex, containing uppercase letters, lowercase letters, numbers, and special characters. The length of the password must be at least 8
* Security measures of the databases will also be implemented.
* Use of HTTPS and SSL certificates for a secured website.
* A secure and reputable web hosting company will be chosen.
* Back up of data on daily basis will be done.
* A security expert will be hired to watch the activity of the website and avoid it from any security threats.

### Additional Software Quality Attributes

The presence of advanced models and tools help us in providing software quality assurance. Our website will be reliable since the reliability depends on the accuracy of the results provided and we will ensure that our model is provided with enough dataset that it will be well trained and provide us with accurate results. Our application will be flexible since it has no restrictions of platforms and with flexibility comes portability. We will also ensure the availability of services and data so that the user does not have bad experience while using the website. Our web content will be compatible with a variety of user agents and it can be ensured by following web standards and testing rigorously. Last but not least, our website will be user friendly and will focus on ease of use and ease of learning by providing attractive and friendly content on the webpages and provide the important features of the website in such a way that they will be automatically highlighted.

## Other Requirements

All requirements have been clearly stated in Final Year report document. No need for additional requirements yet.

# System Design

## Application and Data Architecture

## 

## Figure 2: Entity Relation Diagram

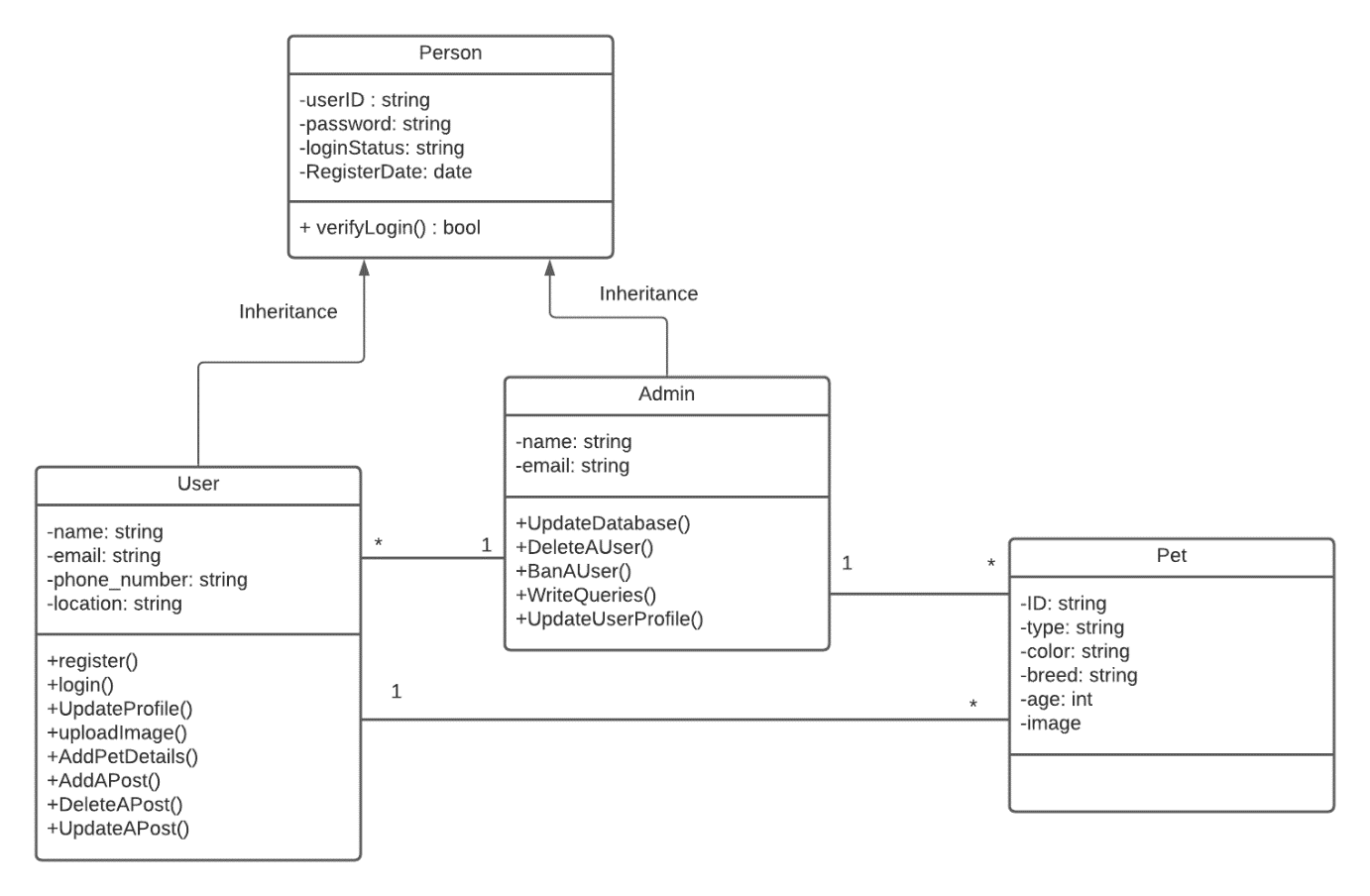
## The entity relationship diagram of our system shows different entities in our systems store information and relationship among these entities.

## Our system has an entity “User” which have attributes name, id, phone number, address and email

## The entity “user” has a “login” option that contains the username and password of the user.

## User has roles that are specific for normal users and privileged for administrators.

* User can create a post by adding details of the pet like its type, location of the lost/found pet, image.
* User views an entity “post” that contain details like image, animal type,status, location of the pet and can make delete/update it too.
* User can also create “account” by the means of selecting a username and password.
* System “notifies” the user if a match is found in the database.

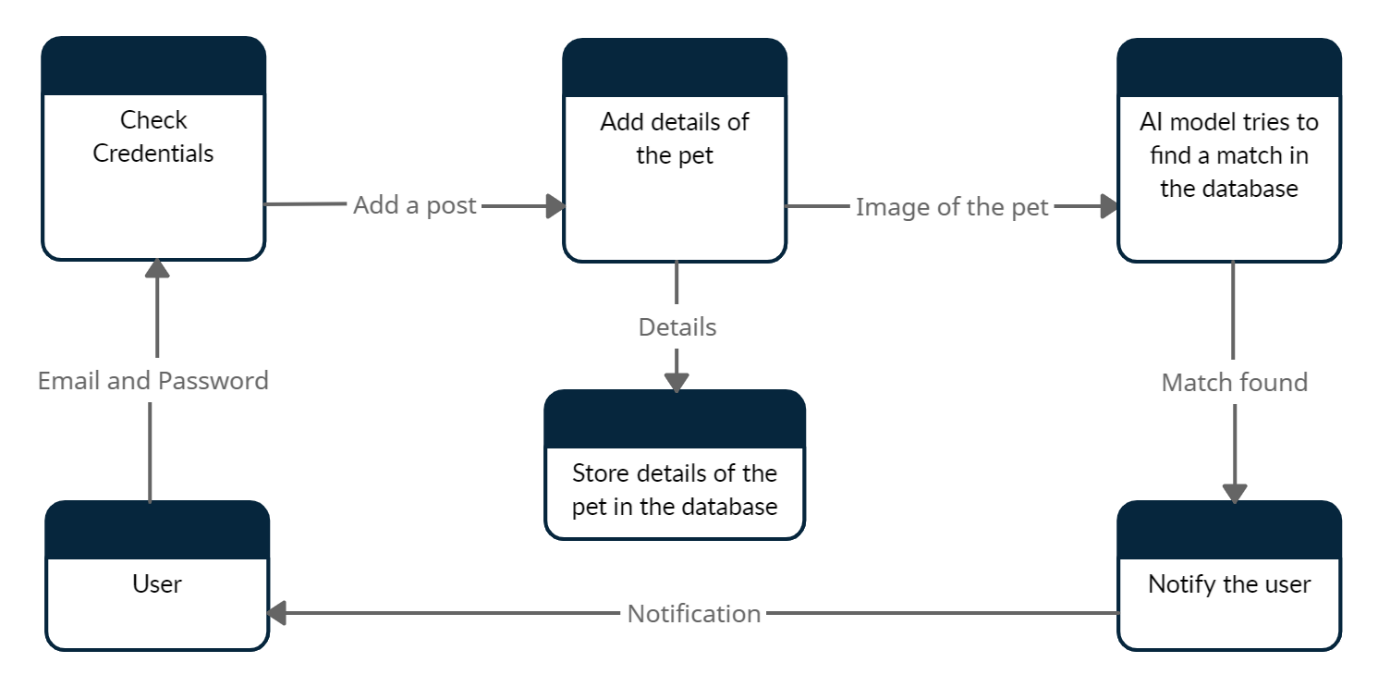


**Figure 3: Class Diagram**



## Figure 4: Component Diagram

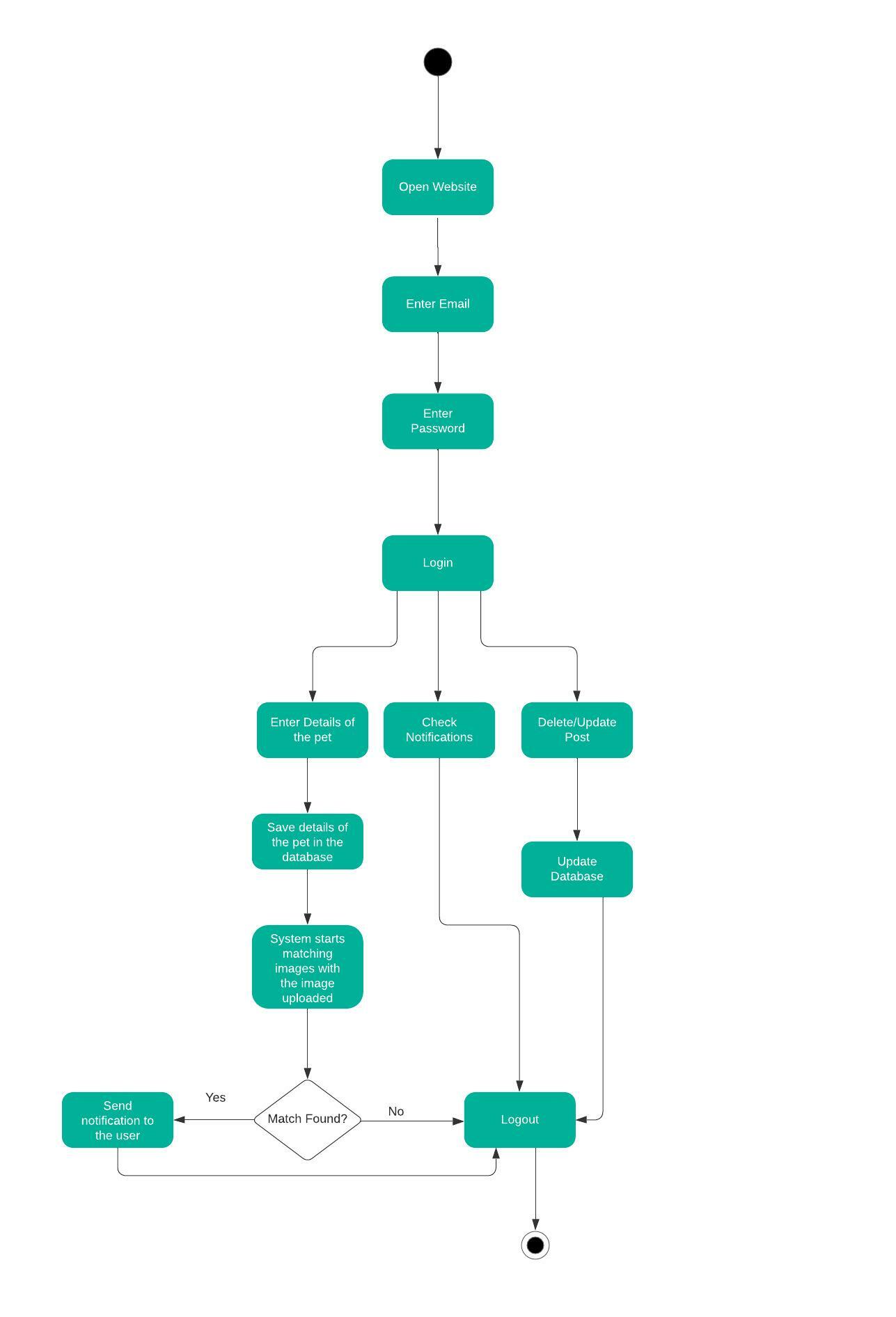
## Component Interactions and Collaborations



**Figure 5: DFD**

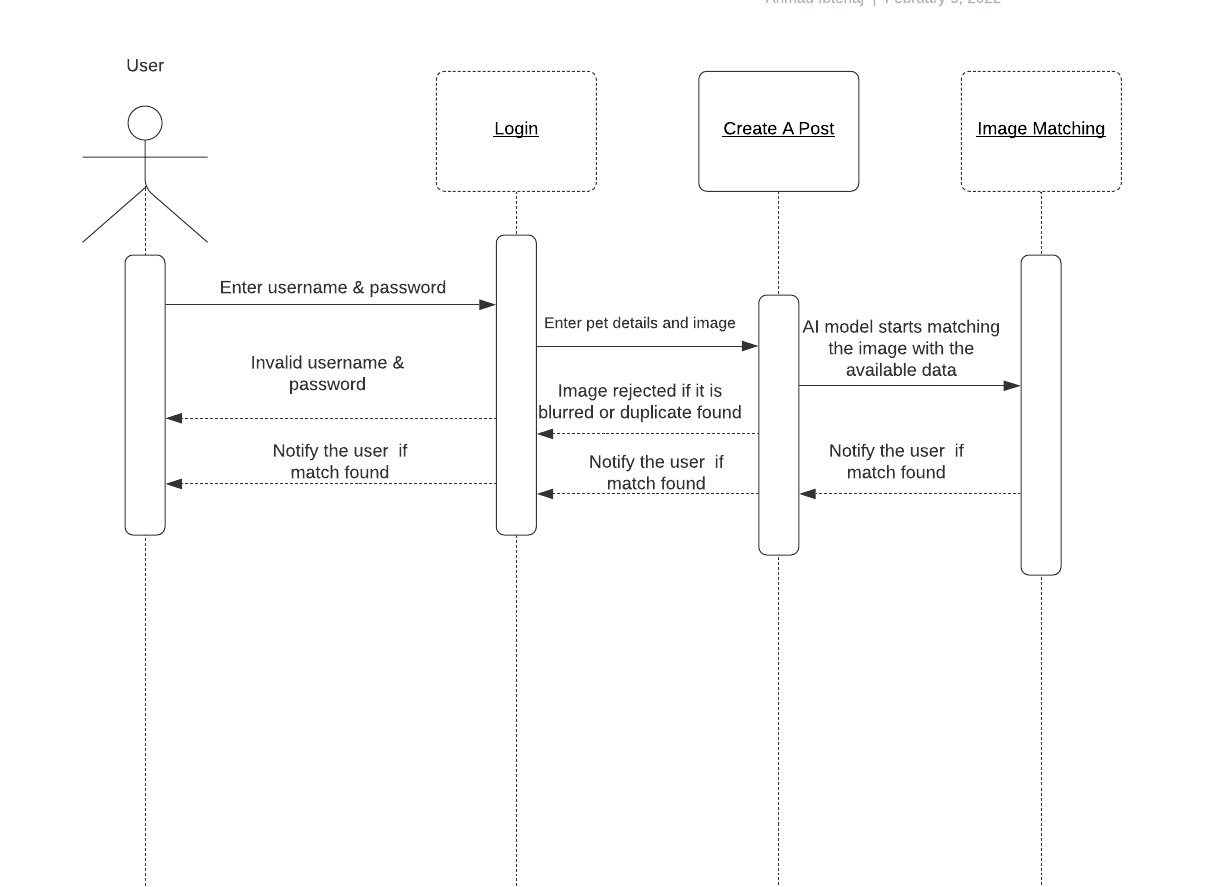
Major components of our system have been shown in the diagram that are:

* **Check Credentials:** User enterthe email and password and the component verifies the credentials.
* **Add Details of the Pet:** This component takes information of the pet like its type, location and the image and creates a post and stores the details in the database and send the image to the AI model for image matching
* **Image Matching Model:** The image is then fed as an input to the image matching model which tries to find a match in the database and send back the image with highest similarity.
* **Notify the User:** If a match is found, then a notification is sent to the user.

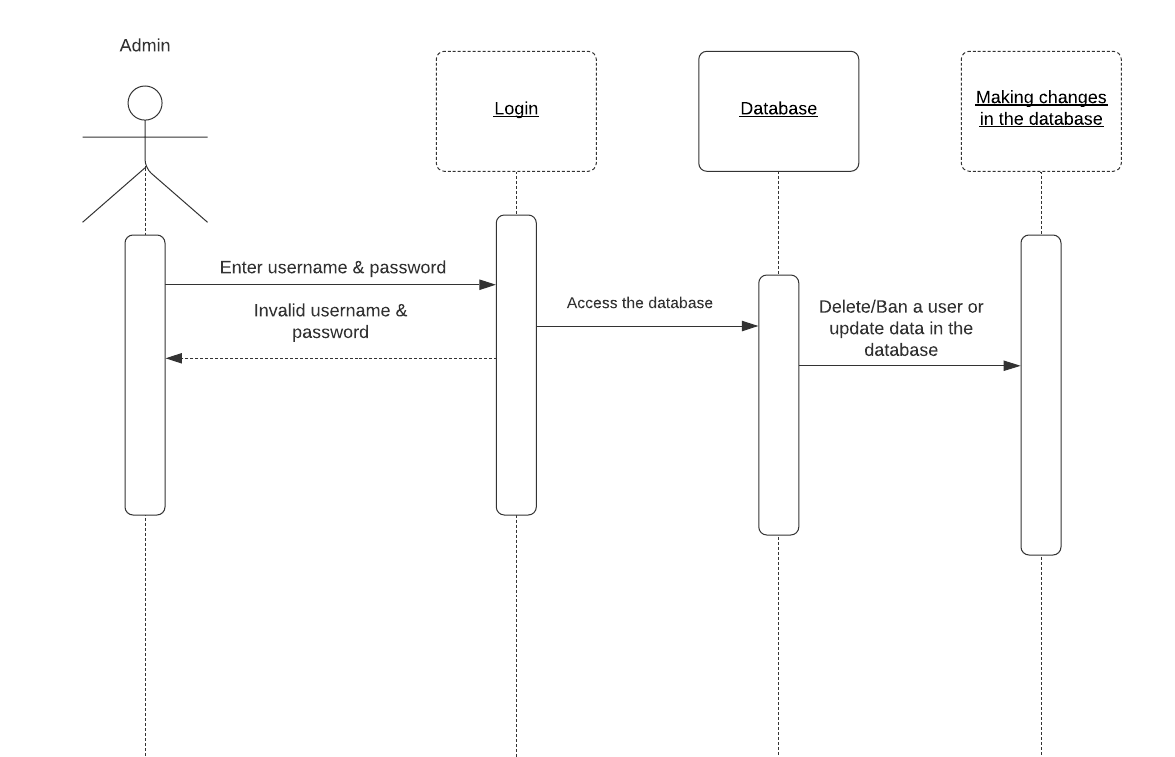


**Figure 6: Activity Diagram**

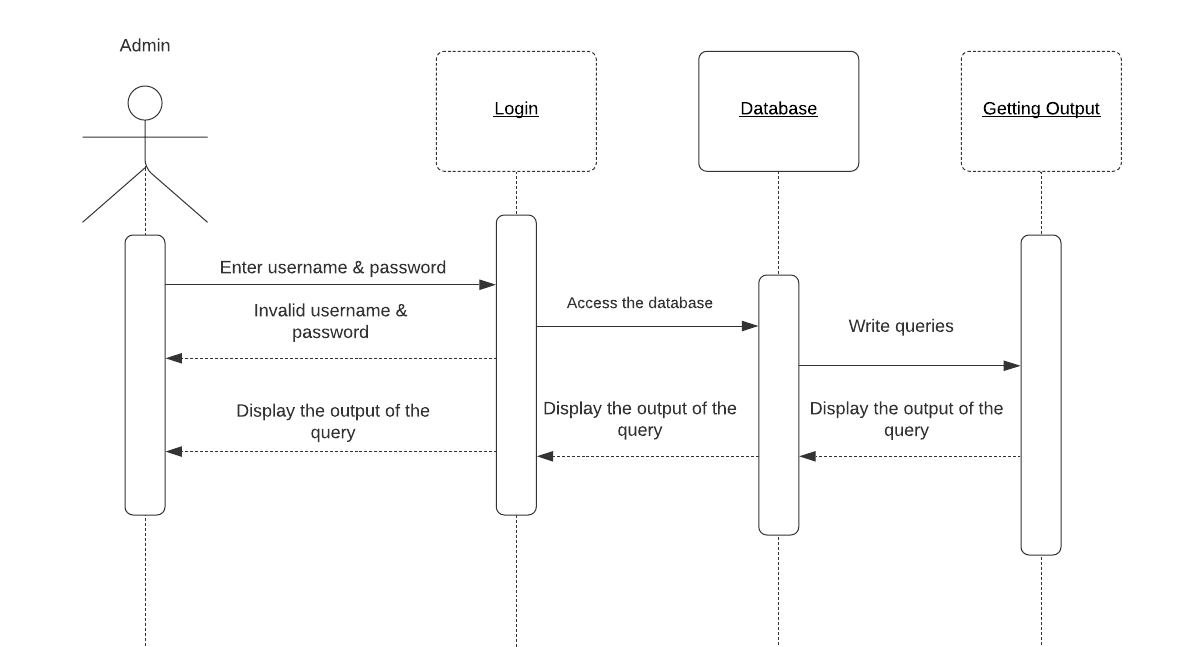
This activity diagram illustrates the flowchart of our system from one activity to the next one.



**Figure 7: Sequence Diagram 1**



**Figure 8: Sequence Diagram 2**



**Figure 9: Sequence Diagram 3**

## System Architecture

This project is a web based application. We have used React for the frontend, Flask for the backend and MongoDB for the database.

When a user opens the web app, he/she has two options i.e., login or signup.

* If user is new, he/she will enter the username, password, email, address and phone number and that will be sent by react to flask where it will be stored in the database.
* If a user already exists, the email and password entered by the user will be sent by react to Flask which will validate the user.

After logging in, the user will click on the “add a post” button in order to create a post. He/she will enter the details of the pet like type, location where it was found/lost and the image of the pet will also be uploaded along with the details. Once the user enters the details of the pet, react will send the details to Flask at the back end. Flask will store the details along with the images in the database and then send the image uploaded by the user to the AI model. The AI model will be provided with all the other images of the lost pets one by one in a loop and the AI model will compare the two images at a given time by extracting their features and provide the output in the form of similarity index. If the similarity is above 90%, then it will consider the image as a match and will send back the image to the Flask. Flask will then create a notification and send back to both the owner and finder of the lost pet and share their details with each other.

**Deep Learning:**

**Siamese Network:**

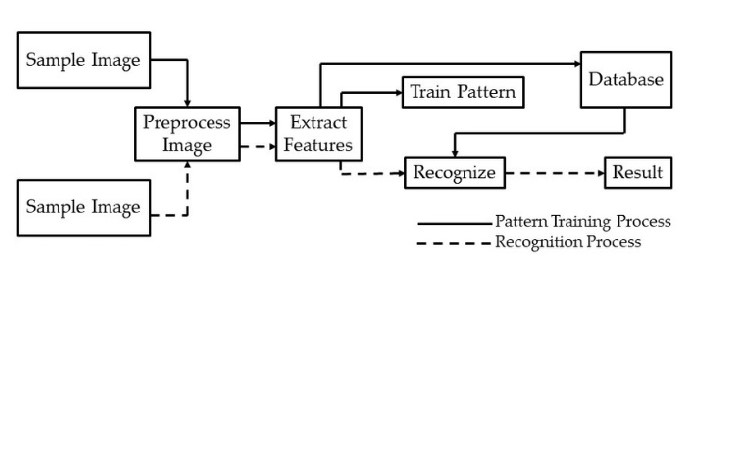
[Siamese Networks](https://en.wikipedia.org/wiki/Siamese_neural_network) are neural networks which share weights between two or more sister networks, each producing embedding vectors of its respective inputs.

In supervised similarity learning, the networks are then trained to maximize the contrast (distance) between embeddings of inputs of different classes, while minimizing the distance between embeddings of similar classes, resulting in embedding spaces that reflect the class segmentation of the training inputs.

**Model Training**

We trained the model to differentiate between pets of different classes. For example, a cat picture needs to be differentiated from a dog picture. Also, one cat type needs to be differentiated from the other cat type. To carry this out, we first trained our model on only one specific type of animal and will expand it with time. We put images of the same cat in one folder and then created pairs by selecting one random image from one folder and other image from another folder and assigned them with class label 0 which means that those are the images of two different cats. Images of the same cats were present in one folder and they were paired with class label 1. We repeated this process for all the images in order to cover all the possible combinations.

Then the dataset was separated into training, validation and testing data. The model preprocesses the images and extract features from within them. The common learning goal is to minimize a distance metric for similar objects and maximize for distinct ones. There are two input layers, each leading to its own network, which produces embeddings. A Lambda layer then merges them using an [Euclidean distance](https://en.wikipedia.org/wiki/Euclidean_distance) and the merged output is fed to the final network. And in this way the model is trained and then its training is tested by giving our model the test data and check its accuracy.



**Figure 10: High Level System Architecture Diagram**

## Architecture Evaluation

* In the machine learning portion of our project where we needed to match images which were viewpoint independent and not some images with a perfect angle, it was a difficult task to choose the model. For this task, we chose Siamese network which is a neural network A Siamese neural network (sometimes called a twin neural network) is an [artificial neural network](https://en.wikipedia.org/wiki/Artificial_neural_network) that uses the same weights while working in tandem on two different input vectors to compute comparable output vectors. Often one of the output vectors is precomputed, thus forming a baseline against which the other output vector is compared. This is similar to comparing [fingerprints](https://en.wikipedia.org/wiki/Fingerprint) but can be described more technically as a distance function for [locality-sensitive hashing](https://en.wikipedia.org/wiki/Locality-sensitive_hashing). The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. For example, given many pictures of cats and dogs it learns distinctive features for each class by itself. CNN is also computationally efficient and Siamese accuracy was above 90% too as compared to the other neural networks available. Some pros of Siamese are:
* It is more robust to class imbalance
* It learns from the semantic similarity
* It is nice to pair with the best classifier

One of the pros of Siamese is that it takes more time to train than normal neural networks.

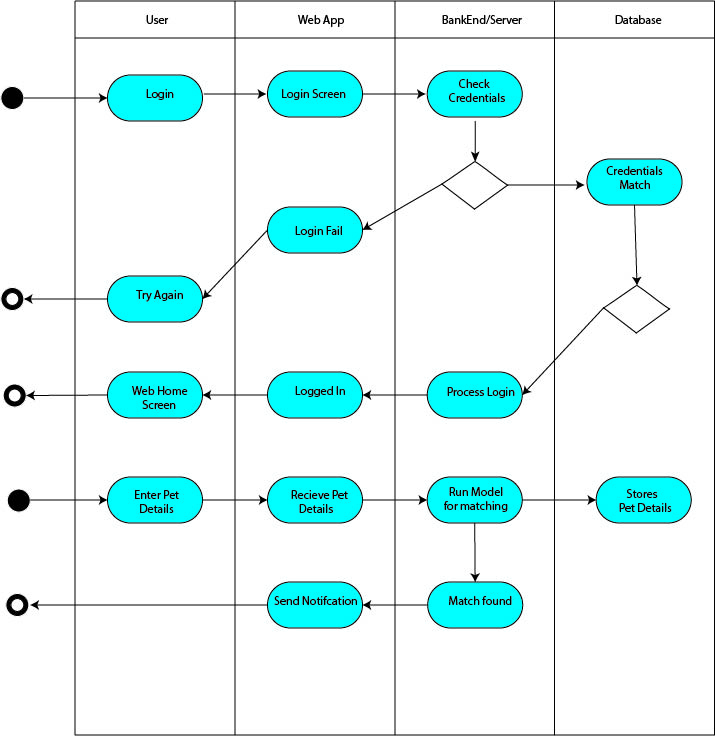
* Talking about the framework, we used Flask for the backend and react for the frontend. Since our model training was in python so it was easy for us to integrate the AI model API with the backend as Flask is a python based framework. Talking about react, so react creates a single page web application so the retrieval and access time is higher as compared to the other frameworks for the frontend thus enhancing the user experience.

## Component-External Entities Interface

The most important aspect of this project is the AI models that we used. For this purpose, we used Siamese network and dense layer in our project. These models were implemented by using TensorFlow machine learning library to train, test and find matches among our pet images data. TensorFlow is a work of Google Brain team and is an open-source library which is used for machine learning and numerical computing. Also, there will be no communication with the external entities as our platform is independent and not linked with any other website.

## Screenshots/Prototype

### Workflow

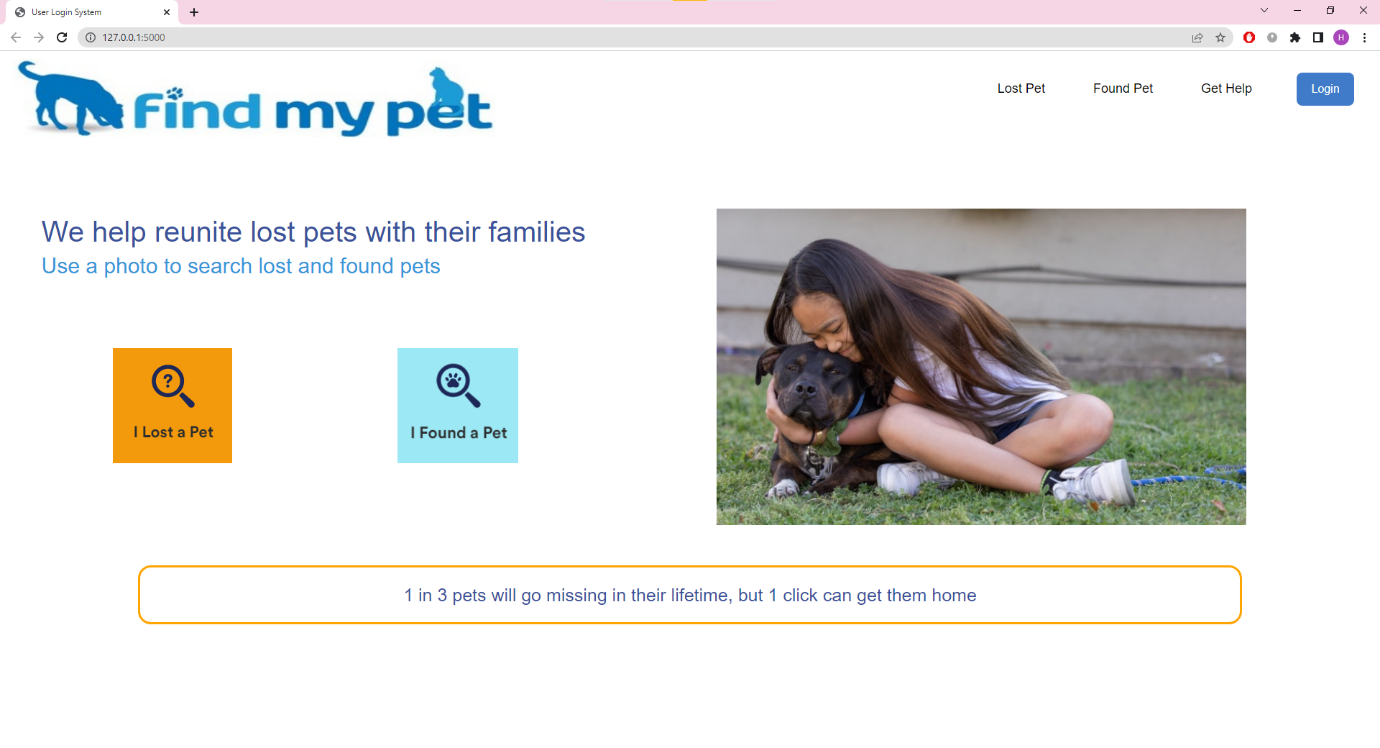


**Figure 11: Swimlane Diagram**

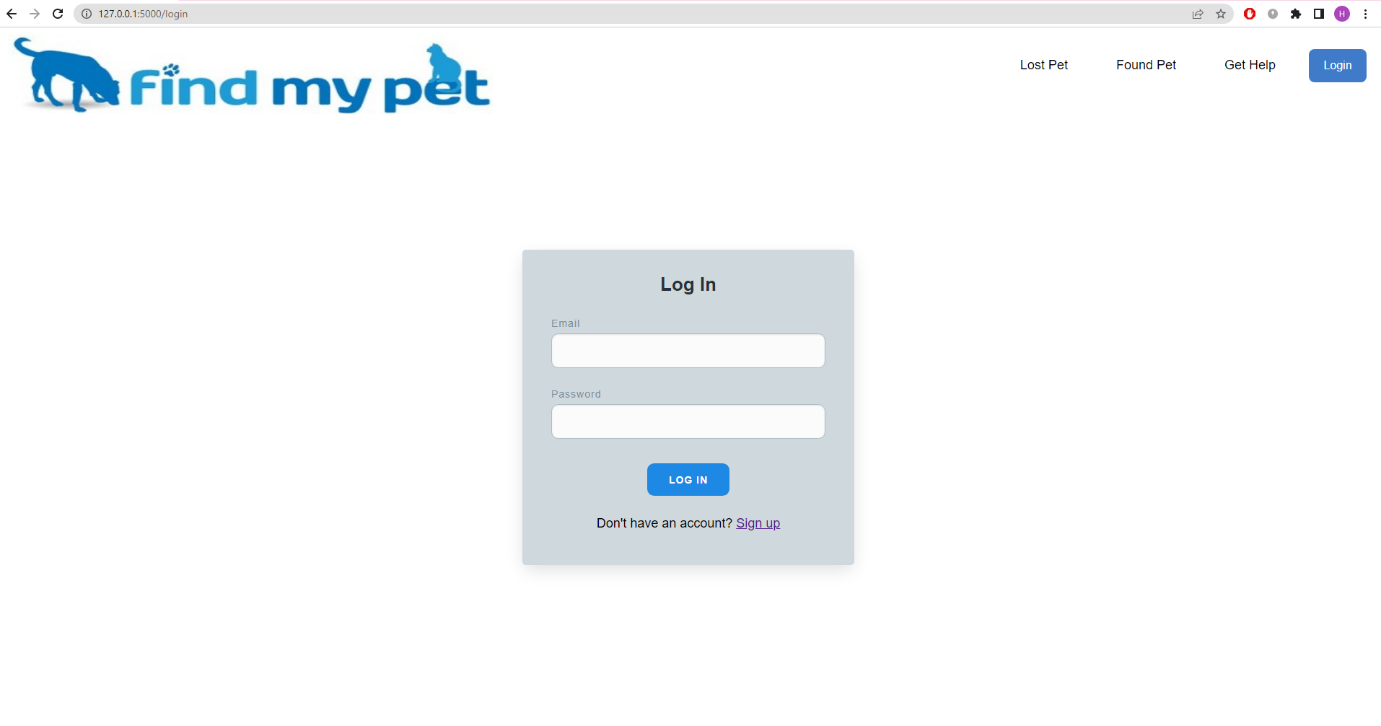
Swimlane Diagram depicts the complete workflow of our system. User, webapp backend server, database, are the main components of our workflow. When user uses web app to log in, backend sever handles the request and using authentication database show a response on the web page. User after logging in use web app to post a picture of the lost/found pet and our server starts finding a match in the database by calling the AI model API. If a match is found, a notification is sent to the user.

### Screens:

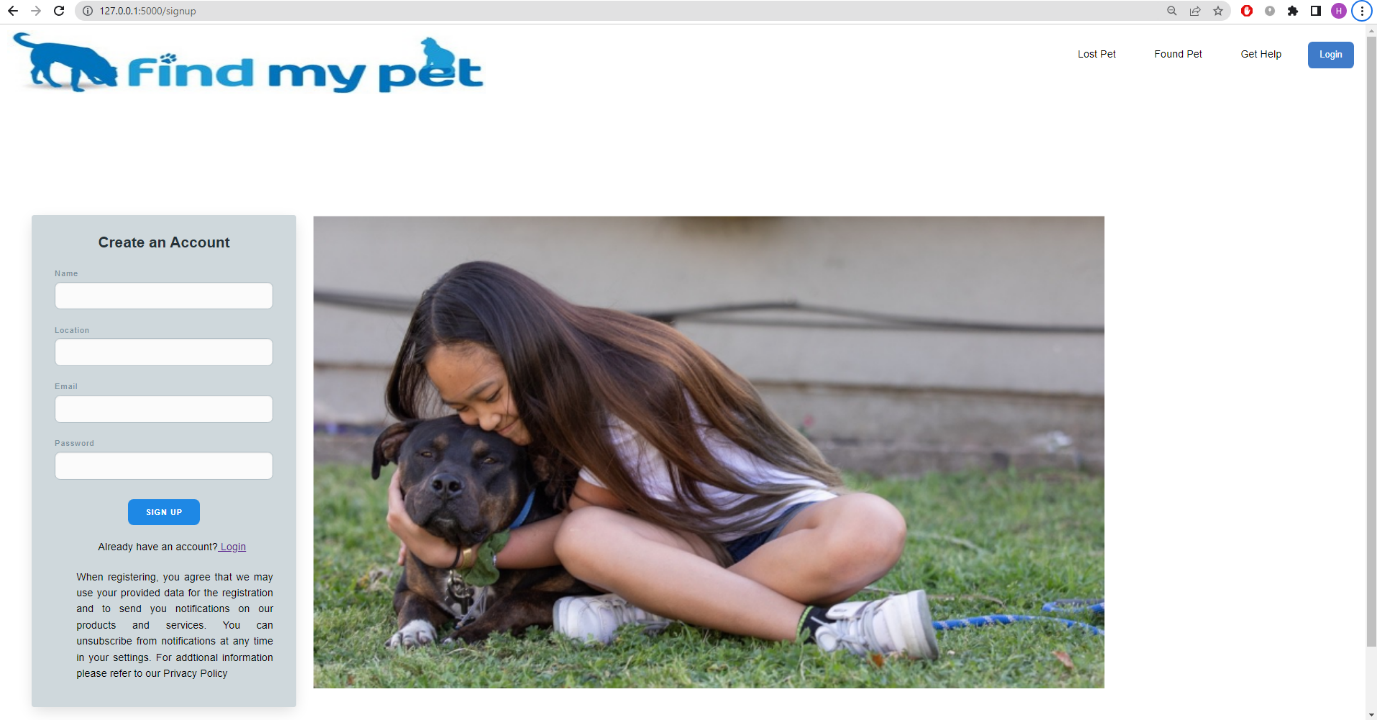
**Screen 1 Landing Page:**



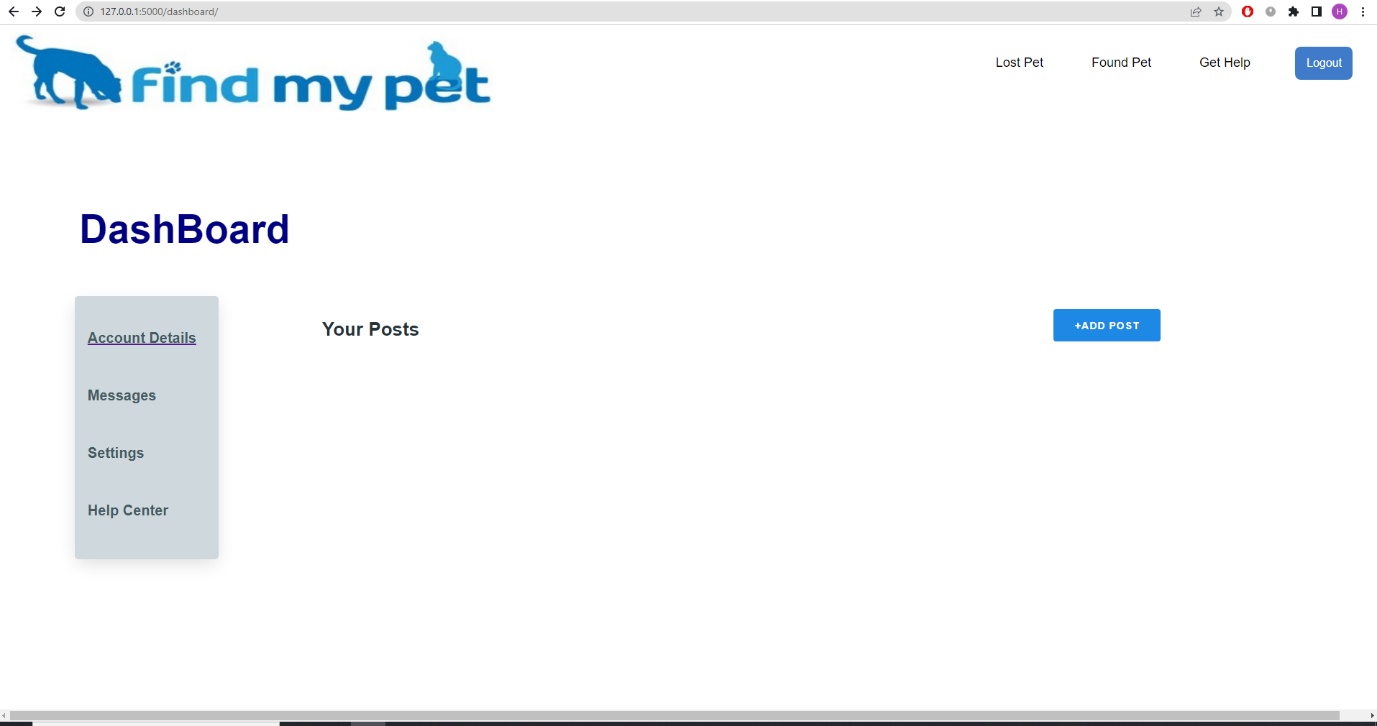
**Screen 2 Login:**



**Screen 3 Signup:**

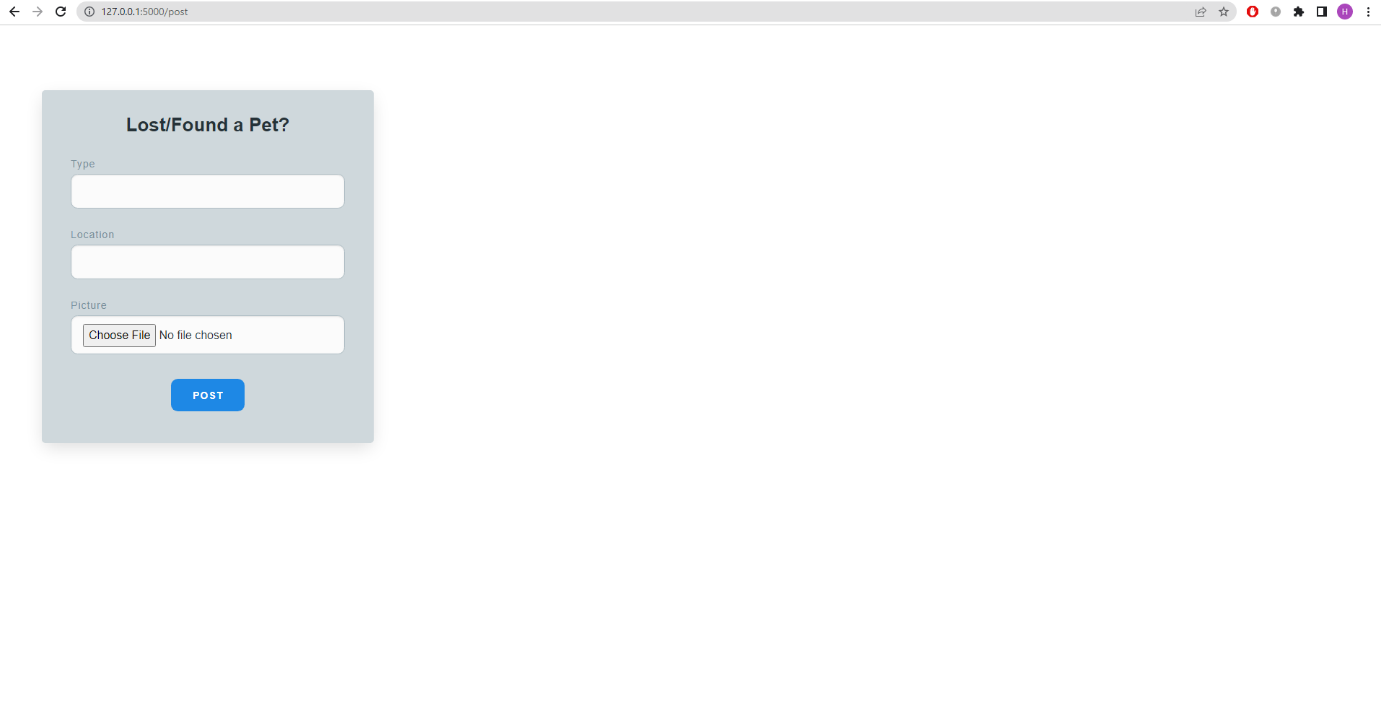


**Screen 4 Dashboard:**



**Screen 5 Add a New Post:**





# Test Specification and Results

## Test Case Specification

Table 7: TC-1

|  |  |
| --- | --- |
| **Identifier** | TC-1 |
| **Related requirements(s)** | UC-1 |
| **Short description** | We will check if the sign up works or not. |
| **Pre-condition(s)** | 1. Internet Connectivity 2. Opening website |
| **Input data** | username and password |
| **Detailed steps** | 1. Enter Username 2. Enter Password |
| **Expected result(s)** | User Signed Up |
| **Post-condition(s)** | Alert showing username signed up and login page opens up successfully. |
| **Actual result(s)** | User Signed Up |
| **Test Case Result** | PASS |

Table 8: TC-2

|  |  |
| --- | --- |
| **Identifier** | TC-2 |
| **Related requirements(s)** | UC-2 |
| **Short description** | Check if the log in works or not. |
| **Pre-condition(s)** | 1. Internet Connectivity 2. Opening website |
| **Input data** | username and password |
| **Detailed steps** | 1. Enter Username 2. Enter Password |
| **Expected result(s)** | User Logged In |
| **Post-condition(s)** | Logging in to the website and the feed page shows up |
| **Actual result(s)** | User Logged In |
| **Test Case Result** | PASS |

Table 9: TC-3

|  |  |
| --- | --- |
| **Identifier** | TC-3 |
| **Related requirements(s)** | UC-3 |
| **Short description** | Check if a post can be created |
| **Pre-condition(s)** | * Internet Connectivity * Opening website * Logging in to the account |
| **Input data** | Pet type, location, image |
| **Detailed steps** | 1. Enter the type of pet 2. Enter the location of the lost/found pet 3. Select and upload the relevant image of the from the computer |
| **Expected result(s)** | Pet details added and image uploaded |
| **Post-condition(s)** | Post shown on the feed of the account |
| **Actual result(s)** | Post shown |
| **Test Case Result** | PASS |

Table 10: TC-4

|  |  |
| --- | --- |
| **Identifier** | TC-4 |
| **Related requirements(s)** | UC-4 |
| **Short description** | Check if the user profile can be updated. |
| **Pre-condition(s)** | * Internet Connectivity * Opening website * User has an already existing account * Logging into the account |
| **Input data** | Name, email, phone number, location. |
| **Detailed steps** | 1. Click on the update button. 2. Enter the new name 3. Enter the new email 4. Enter the new phone number 5. Enter the new location |
| **Expected result(s)** | User profile updated. |
| **Post-condition(s)** | Profile of the user updated. |
| **Actual result(s)** | User profile updated |
| **Test Case Result** | PASS |

Table 11: TC-5

|  |  |
| --- | --- |
| **Identifier** | TC-5 |
| **Related requirements(s)** | UC-5 |
| **Short description** | Check if the post can be updated. |
| **Pre-condition(s)** | * Internet Connectivity * Opening website * Logging into the account * User has an already existing post/posts |
| **Input data** | Type, image, location. |
| **Detailed steps** | 1. Click on the update button with the post 2. Enter the new type 3. Upload the new image 4. Enter the new location |
| **Expected result(s)** | Post updated. |
| **Post-condition(s)** | Post of the user updated. |
| **Actual result(s)** | Post updated |
| **Test Case Result** | PASS |

Table 12: TC-6

|  |  |
| --- | --- |
| **Identifier** | TC-6 |
| **Related requirements(s)** | UC-6 |
| **Short description** | Check if the post can be deleted |
| **Pre-condition(s)** | * Internet Connectivity * Opening website * Logging into the account * User has an already existing post/posts |
| **Input data** | Null |
| **Detailed steps** | Click on the delete button with the post |
| **Expected result(s)** | Post deleted. |
| **Post-condition(s)** | Post of the user deleted |
| **Actual result(s)** | Post deleted. |
| **Test Case Result** | PASS |

## 

## Summary of Test Results

Table 13: Summary of All Test Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module Name** | Test cases run | Number of defects found | Number of defects corrected so far | Number of defects still need to be corrected |
| **Module 1 (Sign Up)** | TC1 | 0 | 0 | 0 |
| **Module 2 (Log In)** | TC2 | 0 | 0 | 0 |
| **Module 3 (Make a post)** | TC3 | 0 | 0 | 0 |
| **Module 4 (Update Profile)** | TC4 | 0 | 0 | 0 |
| **Module 5 (Update Post)** | TC5 | 0 |  | 0 |
| **Module 6 (Delete Post)** | TC6 | 0 | 0 | 0 |
| **Complete System** | TC1, TC2, TC3, TC4, TC5, TC6 | 0 | 0 | 0 |

# Conclusion and Future Work

## Project summary

Losing your pet has become one of major problems in the world and the techniques applied to find them are very time consuming or of no use. Since this era is technology based and machines are artificially intelligent so we have tried to cater this problem using artificial intelligence. Now people can use this lost pet finder platform to find their pets just by uploading the details and viewpoint independent images of their pets. The viewpoint independent concept plays a vital role because many people of old generation don’t know how to take a proper picture so, if the image is not taken with a perfect angle, it’s fine because this platform is independent of such a problem. Now people can use this platform to find their beloved pets in a very organized and timely manner as it is hassle free and very convenient to use. Also, the location based searching makes the finding of the pet more optimized.

## Problems faced and lessons learned

Problems faced were:

* Finding correct and large amount of data to train AI models.
* Creating dependable image matching in an angle and viewpoint-independent manner is challenging, especially in the case of pets because it can be difficult to tell same breeds apart.
* Selecting the framework to work on for the development of the website.
* Incorporating AI models through Flask at the back end of the application was a challenging task.
* Training and testing large amount of data on google collab with limited amount of data usage per day.

## Future work

* This lost pet finder platform is currently a web application and there is no mobile application for it. The future works will include the formation of iOS and android mobile application for this platform.
* NGOs will be involved and the feature of nearest pet care house will be added where people can give away their pets to be taken care of.
* Sale/ Purchase of pets will also be done through this platform in the future.

# References

1. Lord L.K., Wittum T.E., Ferketich A.K., Funk J.A., Rajala-Schultz P. Search methods that people use to find owners of lost pets. J. Am. Vet. Med. Assoc. 2007;230:1835–1840. doi: 10.2460/javma.230.12.1835.

2. 7 Great Organizations Helping Bring Lost Pets Home. PetHub. (2021, November 2). Retrieved November 28, 2021, from https://www.pethub.com/article/7-great-organizations-helping-bring-lost-pets-home.

3. Lost pet report. website. (n.d.). Retrieved November 28, 2021, from https://www.lostfoundpets941.com/lost-pet-report.

4. Adopt a new friend today! City of Los Angeles Department of Animal Services Banner. (n.d.). Retrieved November 28, 2021, from https://www.laanimalservices.com/about-animals/lost-pet/.

# Glossary

**Analysis:** A detailed examination of anything complex in order to understand its nature

**Deep Learning:** Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning.

**Viewpoint Independent:** Something that can identify an object based on its structural description of its parts and the relations of those parts and can identify an object from any viewpoint as long as we see it key features.

**Computer Vision:** Computer vision is a field of artificial intelligence that trains computers to interpret and understand the visual world. Using digital images from cameras and videos and deep learning models, machines can accurately identify and classify objects — and then react to what they “see.”

**Authentication:** The process or action of proving or showing something to be true, genuine, or valid.

**Sentiment:** A view or opinion that is held or expressed.

**Digital Image Processing:** Digital image processing deals with manipulation of digital images through a digital computer. It is a subfield of signals and systems but focus particularly on images. DIP focuses on developing a computer system that is able to perform processing on an image.

**Time-intensive:**  Liable to absorb or demand a great deal of time.

**Futile:** Incapable of producing any useful result; pointless.

# Deployment/Installation Guide

Our project is a web-based application that can be accessed through a stable internet connection once it is deployed on a web hosting service.

# User Manual

Following steps should be performed by the user:

* Open website by entering the correct domain name/url.
* If using the app for the first time, choose sign up option.
* Signup by entering username, password, email, phone number, location.
* If a user already exists, enter username and password to login and click on the login button.
* Click on the “add new post” button and enter details of the pet and upload image by browsing through your pc and then click on the create button in order to create a new post.
* Click on the “update post” button and enter the new details of the pet and click on the “update” button in order to update the post.
* Click on the “delete post” button and in order to delete the post.
* Click on the “update profile button” and the enter the new details and click on the “update” button in order to update the profile.
* Click on the “messages” option on the left side of the home screen in order to check notifications.

# Student Information Sheet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Roll No | Name | Email Address (FC College) | Frequently Checked Email Address | Personal Cell Phone Number |
| 22-10588 | Ahmad Ibtehaj | 22-10588@formanite.fccollege.edu.pk | ahmadibtehaj456@gmail.com | 0333-1475748 |
| 22-10208 | Hamza Khan | 22-10208@formanite.fccollege.edu.pk | hamza.khan96021@gmail.com | 03110141009 |
| 22-10658 | Rabia Tahir | 22-10658@formanite.fccollege.edu.pk | 22-10658@formanite.fccollege.edu.pk | 03116747071 |

# Plagiarism Free Certificate

This is to certify that, I am \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ S/D/o \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, group leader of FYP under registration no \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ at Computer Science Department, Forman Christian College (A Chartered University), Lahore. I declare that my Final year project report is checked by my supervisor and the similarity index is \_\_\_\_\_\_\_\_% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix F. To the best of my knowledge and belief, the report contains no material previously published or written by another person except where due reference is made in the report itself.

Date: \_\_\_\_\_\_\_\_\_\_\_\_ Name of Group Leader: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Supervisor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Co-Supervisor (if any):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Designation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Designation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Senior Project Management Committee Representative: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Plagiarism Report