Chapter 2

2.1 Introduction to Analysis

Analysis is a methodical study of information by breaking it into simpler form which later becomes a foundation to successful problem solving and decision making. It is very crucial in the determination of success of the project. In order to ensure that our system fulfills the aims of the project, we need to understand the problem statement thoroughly and analysis helps us to understand it. Analysis also assists us to be clarified about the unclear and unspecified requirements. Additionally, it aids in controlling non-mandatory features of the system.

2.2 Analysis Methodology

The systematic and well-structured process for requirements analysis can be understood as analysis methodology. Electing a right methodology is very important for a system to reach its success. It is done on the basis of system's complexity, technical requirements, organizational requirements and users' requirements. A number of methods for analysis can be found, which are, Hard Systems Methodology, Soft Systems Methodology, Combined methodology, Object-oriented Methodology, etc.

Among all of these options, I have chosen Hard Systems methodology for my online pet adoption system. Hard systems methodology is a highly structured approach that strictly follows a logical order of rules, standards and guidelines. The reasons of choosing this methodology are as follows:

i. Ensures rigorous planning and scheduling.
ii. Makes it feasible to measure progress in reference to the objectives planned.
iii. Ensures that each step is completed before moving to the next one.
iv. Suitable for small to large organization systems.

Since it follows Structured Systems Analysis and Design (SSADM) method, we will need to design a Data Flow Diagram (DFD).

DFD (Data Flow Diagram): A DFD (Data Flow Diagram) represents the flow of information within a system with the help of standard symbols like arrows, circles, rectangles, text, etc. It helps in modelling a system and gives an overview of the functions of the system.

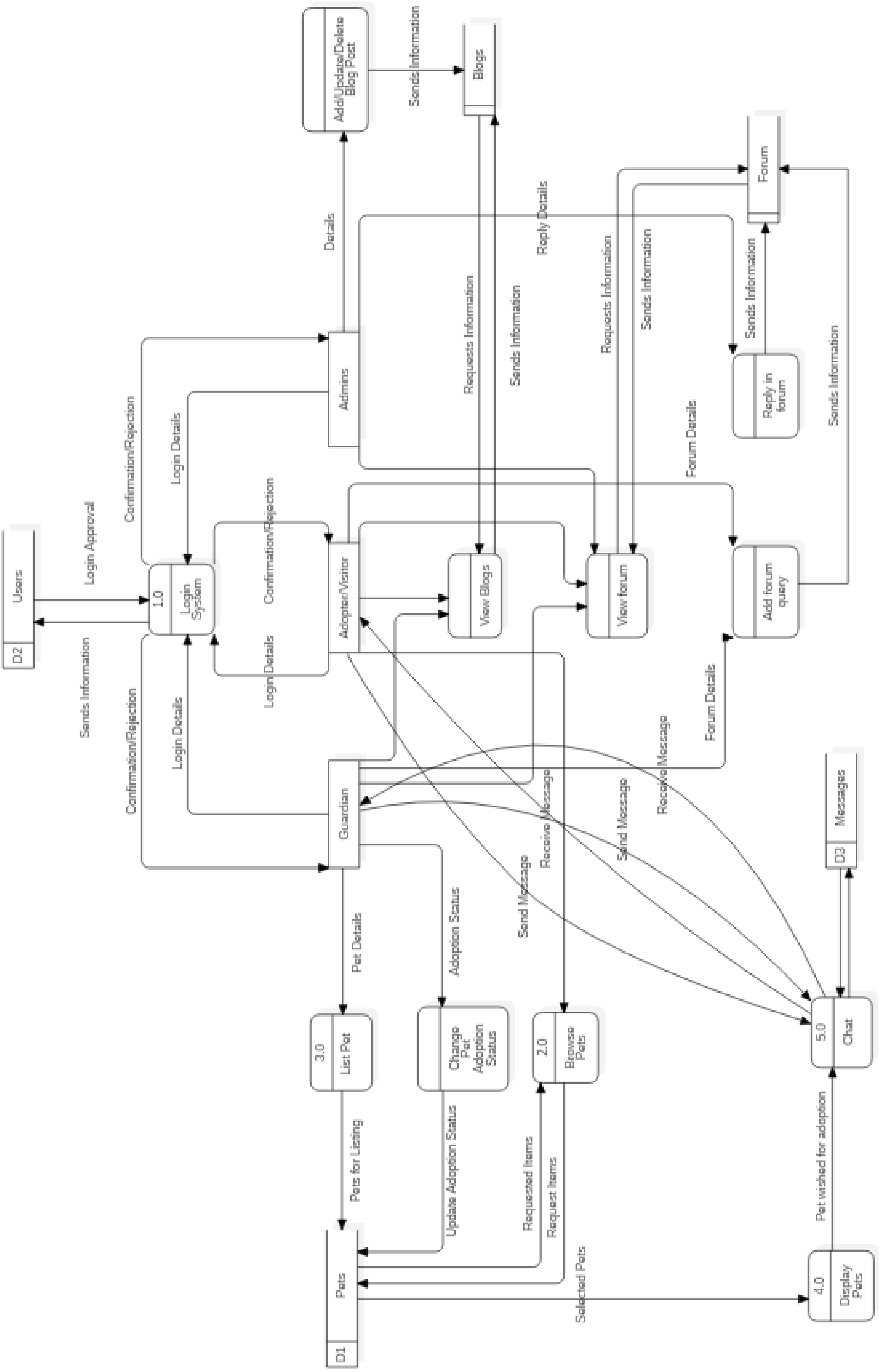


Figure 1: Data Flow Diagram (DFD) of the system

2.3 Feasibility Study

The process of overviewing the possibility of successful implementation of a system while focusing on identifying probable complications is known as feasibility study. The study tries to answer two major questions; Will the system work in practicality, and whether the system should be initiated. The major areas covered by feasibility study are discussed below:

i. Technical feasibility:

Does the project have all the required technical resources? Technical feasibility focuses on the accessibility of the hardware, software and technologies required in the development of the system.

The system I am going to develop is adequate in all the necessary hardware, software and technologies which proves that my system is technically feasible

ii. Operational feasibility:

How likely is the proposed system going to cover the problem statement and fulfill the aims of the project? Operational feasibility aids us in predicting if the system meets the clients' requirements or not.

The website will be thoroughly planned and will be implemented carefully too. It will ensure that the project will be operationally feasible.

iii. Market feasibility:

Does the project have the potentiality to operate well in the market in future? Market feasibility predicts the present and future market status of the system or project. It gives an overview of the type of individuals that will use the system, the impact it will make and other information for marketing purposes.

Since the website is first of its kind in Nepal, there is no doubt that the website will not be feasible in market domain.

iv. Legal feasibility:

Does the project adhere within the boundaries of the legal requirements? Legal feasibility makes sure that the project follows the laws like copyright law, data protection law, etc. and does not fall outside the boundaries in any way possible.

The website does not include contents of legal issues. Hence it will be legally feasible.

iv. Financial feasibility:

Financial feasibility is performed in order to find the best way possible to develop the system in such a way that the cost is low and the benefit is high.

As the system is being developed for academic purpose, there is no noticeable cost in developing the website. Hence, the system is financially feasible too.

v. Behavioral feasibility:

Behavioral feasibility monitors the reaction of users towards the system. It ensures that the system is easy-to-use and meets each of the requirements of the users.

The website will be user-centered so that it will be easy to use for the users. It can be said that it is feasible in behavior too.

2.4 SRS (Software Requirement Specification)

The documentation of all the functional and non-functional requirements of the system to be designed is known as SRS (Software Requirement Specification). It consists of hardware and software requirements as well.

2.4.1 Functional Requirements

Functional requirements are those features or functions that need to be implemented in the system. These are the major requirements of the project and without them, the project is incomplete. I have listed all the functional requirements of my project in the table below:

FR ID	Functional Requirements	Data Required	Rational Motive	Dependency
FR1	Admin Login	Username/Email and Password	To allow only verified administrators to access the system which leads in preservation of confidentiality, integrity and authenticity.	N/A
FR2	View, Create, Update or Delete Blogs	Blog title and content	To allow administrators to create, modify or delete blog posts.	FR1
FR3	View and Reply in forum posts	Reply text	To allow administrators to view queries and reply to them in the forum.	FR1
FR4	User Registration	First Name, Last Name, Email ID, Phone Number, Username, Password	To store users' data for future use, including their login credentials.	N/A
FR5	User Login	Username/Email and Password	To allow necessary access only to the verified users.	FR4
FR6	List a pet	Pet Details	To list the in the pets browsing section.	FR5
FR7	Edit or Delete listed pet	Pet Details	To modify data about the pets or remove it from the database including the browsing section.	FR6
FR8	View message and reply	Reply text	To facilitate communication system between guardians and adopters.	FR5
FR9	Update pet as 'adopted'		To remove the pet from the browsing section so that the pet can no longer be available to other adopters.	FR8

FR10	Browse Pets according to various filters	Filter details that consist age, gender, size, breed, etc.	To help potential adopters to find the most suitable pets according to their needs.	N/A
FR11	Message the guardian of the pet	Message Text	To facilitate communication system between guardians and adopters.	FR5, FR10
FR12	Share a successful adoption story	Text	To encourage other users to encourage adoption.	FR5, FR11, FR9
FR13	View and Reply in forum posts	Reply text	To allow administrators to view queries and reply to them in the forum.	FR5
FR14	Post a question in forum	Question	To post a query about the website or anything related to it.	FR5
FR15	View Blogs		To allow users to get updated about various news and articles.	N/A
FR16	Manage Profile	First Name, Last Name, Email ID, Phone Number, Username, Password	To allow users to change their details according to their wish.	FR4
FR17	Recover Password	Email or Phone Number	To allow users to recover their password if forgotten	N/A

2.4.2 Non-Functional Requirements

Non-functional requirements are those requirements which determine the quality of the system. They are not crucial for the system to be deployed, but they are the factors that plays a very vital role in making the system faster, reliable and easily maintainable. I have listed all the non-functional requirements of my project in the table below:

ID	Non-Functional Requirements	Description
NFR1	Performance	The system to be developed must be fast and effective in delivering the contents to users. It must respond to user inputs instantly.
NFR2	Scalability	The system should be able to handle a large number of users at a single time. It should be uptime and should not lag significantly while doing so.
NFR3	Efficiency	The system should function properly as it was designed to function. The users should have the feeling of coherency and logicality when using the website.
NFR4	Reliability	The website should be up and running while functioning as designed to without experiencing any failures.
NFR5	Maintainability	The system should be easily maintainable if any bugs need to be repaired or a new feature need to be added.
NFR6	Confidentiality	The system should be able to protect sensitive data and allow only authorized access. It can be implemented using encryption algorithms, enforcing strong password policy, controlling user access, etc.
NFR7	Integrity	Data should be kept authentic, accurate and not corrupted. Enrolling a proper hosting plan, deploying SSL certificates are some techniques to preserve integrity.
NFR8	Availability	The system should be available at all times excluding remarkably short downtimes while undergoing maintenance.
NFR9	Usability	The system should be easy to use and learn for the users. It should also be usable across different devices and platforms. It can be done by implementing responsiveness, using attractive colors, pictures and fonts, etc.
NFR10	Accessibility	The system should be accessible by the users with a broad range of abilities. It can be achieved by implementing some techniques like keyboard friendly, proper use of colors, giving alternative texts to images, etc.

2.4.3 MoSCoW Prioritization

It is very crucial to understand the priority of tasks to be completed in a project. MoSCoW is a technique for prioritizing the tasks. The letters stand for:

M - Must Have: Non-negotiable requirements of the project.

S - Should Have: Not vital requirements but carrying significant value for the project.

C - Could Have: Desirable requirements which are provided only in best case scenarios.

W – Will Not Have: Expected requirements which will be provided in later dates.

The priority table of all the functions of my system are shown below:

ID	Functional Requirements	MoSCoW
FR1	Admin Login	Must Have
FR2	View, Create, Update or Delete Blogs	Could Have
FR3	View and Reply in forum posts	Should Have
FR4	User Registration	Must Have
FR5	User Login	Must Have
FR6	List a pet	Must Have
FR7	Edit or Delete listed pet	Should Have
FR8	View message and reply	Must Have
FR9	Update pet as ‘adopted’	Must Have
FR10	Browse Pets according to various filters	Should Have
FR11	Message the guardian of the pet	Must Have
FR12	Share a successful adoption story	Should Have
FR13	View and Reply in forum posts	Should Have
FR14	Post a question in forum	Should Have
FR15	View Blogs	Should Have
FR16	Manage Profile	Should Have
FR17	Recover Password	Could Have

ID	Non-Functional Requirements	MoSCoW
NFR1	Performance	Must Have
NFR2	Scalability	Could Have
NFR3	Efficiency	Should Have
NFR4	Reliability	Must Have
NFR5	Maintainability	Could Have
NFR6	Confidentiality	Must Have
NFR7	Integrity	Should Have
NFR8	Availability	Must Have
NFR9	Usability	Should Have
NFR10	Accessibility	Could Have

2.4.4 Hardware/Software Specifications

These are the pre-requisites that defines the hardware components and software resources for a system to be used efficiently. The required hardware and software requirements for online pet adoption system are:

Hardware Specifications:

S.N.	Hardware Type	Specification
1.	Processor	Intel Pentium 4 or later
2.	RAM	2 GM minimum (4 GB recommended)
3.	Hard Disk	10 GB minimum
4.	Internet Connection	High-speed internet connection with a speed of 4 Mbps or higher
5.	Peripherals	Keyboard and a mouse

Software Specifications:

S.N.	Software Type	Specification
1.	Operating System	Windows 7 or later (For Windows Users) Mac OS X Yosemite 10.10 or later (For Mac Users)
2.	Browser	Any
3.	Front-end	bootstrap 4.0.0, jQuery 3.3.1
4.	Back-end	XAMPP 3.2.2, PHP, MySQL

2.5 Use Case Diagram

Use Case Diagram is a representation of functionalities available in the system. Components like actors, use cases, relationships, and boundary boxes are used to show how the system works.

a. **Actors:** Individuals or group of individuals using the system represented by stick figures.
b. **Use Cases:** Processes of the system represented by ovals.
c. **Relationships:** Relations between users and processes.
d. **Boundary Boxes:** Represents a system.

i. Use case of Users

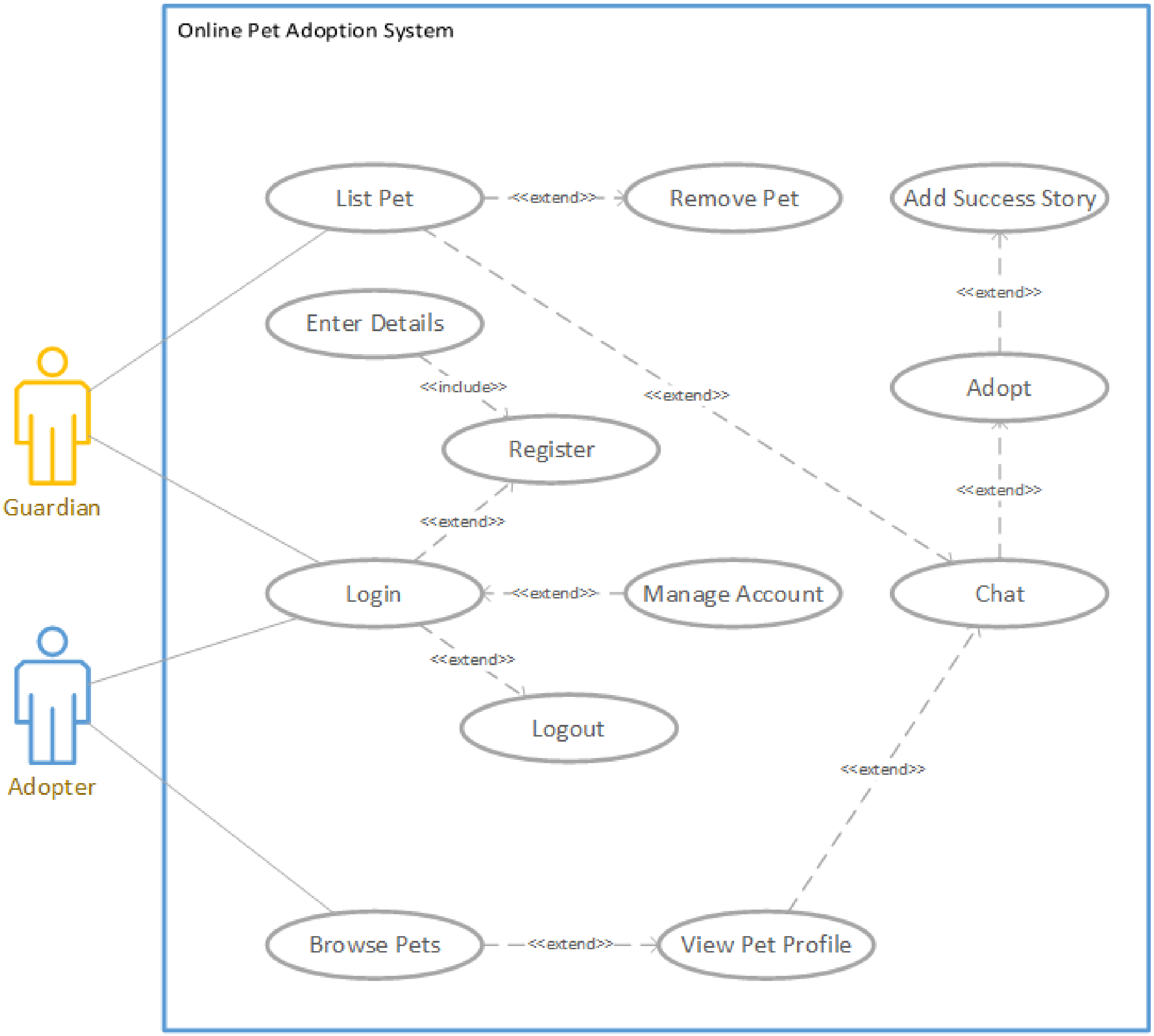


Figure 2: Use Case Diagram of users in Online Pet Adoption System.

In reference to the Use Case Diagram, the functions that can be performed by the users are:

1. Login into the system.
2. If not registered, the user can register using his/her details.
3. After logged in, the use can manage his/her account which allows him/her to change his/her information and even delete their account.
4. The user can logout.
5. Guardians can list their pets for adoption.
6. Guardians can remove their pet if wanted.
7. Guardians can chat with the Adopters and provide further information which can lead to the adoption. After adoption, the guardian can update the adoption status of the pet.
8. Adopter can browse pets.
9. Adopter can view specific pet's profile if he/she is interested.
10. Adopter can then chat with Guardians for further information.
11. Adopter can adopt the pet.
12. Adopter can share an adoption success story for other users to view.

ii. **Use Case of Admins**

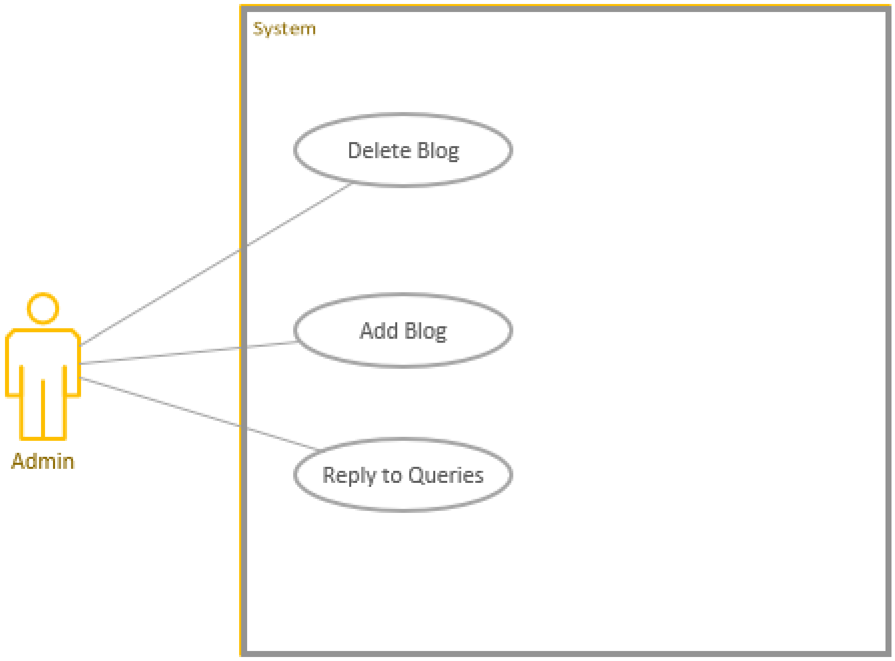


Figure 3:Use Case Diagram of admins in Online Pet Adoption System.

2.6 Natural Language Analysis (NLA)

Natural Language Analysis is a method of separating nouns, verbs and adjectives from the scenario of the system to be developed. These separated nouns, verbs and adjectives are later used as class, methods and attributes respectively.

Scenario

A web-platform needs to be created in order to act as an intermediary among people willing to let their pets for adoption or adopt one. It will allow the guardians of pets to list their pets for adoption with all the details including photos, name, breed, age, gender, size and color.

The website will need to have the following features:

1. Allow users to register by providing their personal details.
2. Allow users to modify their personal details.
3. Implement a login authentication system for the users and admin.
4. Allow guardians to list their pets for adoptions.
5. Allow guardians to provide additional details of the pets.
6. Allow guardians to change adoption status or delete pets.
7. Help adopters to find the most suitable pet based on their preferences.
8. Allow message service between adopters and guardians in case of further information and adoption.
9. Display successful adoption stories to other users.
10. Provide information about pet care to users.
11. Include a blog section where articles can be posted, modified and deleted by admins.
12. Include a forum section where the users can ask their queries while other users or admins can reply to it.

List of Nouns, Verbs and Adjectives identified:

Nouns	Verbs	Adjectives
Web-platform, intermediary, people, pets, guardians, website, adopters, message-service, information, display, adoption-stories, users, blog, articles, admins, forum, queries, admins, login-system, personal details	Created, act, let, adopt, allow, list, need, have, help, find, provide, post, modify, delete, include, ask, reply, register, login, implement, change	Photo, name, breed, age, gender, size, color, personal, suitable, successful

List of Candidate Classes, Methods and Attributes:

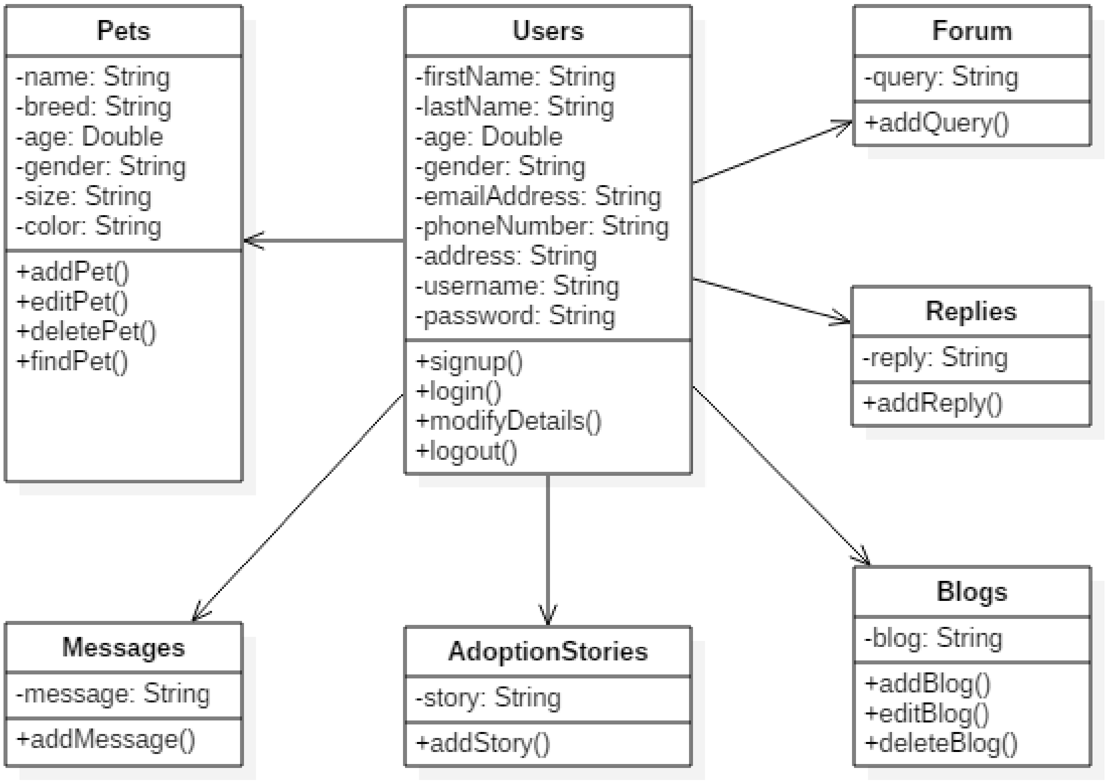
Candidate Classes	Candidate Methods	Candidate Attributes
Users, Pets, Message, Adoption Stories, Blog, Forum	Register User, Login, Add Pet, Edit Pet Details, Delete Pet, Find Pet, Send Message, Add Adoption Story, Add Blog, Edit Blog, Delete Blog, Add Query, Reply to Query	Photo, Name, Breed, Age, Gender, Size, Color,

Elected Classes, Methods and Attributes:

S.N.	Nouns	Verbs	Adjectives
1.	Users	Signup Login Modify details Logout	First Name Last Name Age Gender Email Address Phone Number Address Username Password
2.	Pets	Add Pet Edit Pet Details Delete Pet Find Pet	Name Photo Breed Age Gender Size Color
3.	Messages	Add Message	Message
4.	Adoption Stories	Add Adoption Story	Story
5.	Blogs	Add Blog Edit Blog Delete Blog	Blog
6.	Forum	Add Query	Query
7.	Replies	Add Reply	Reply

Initial Class Diagram

Class Diagram is a static diagram that depicts the structure of a system with the help of its classes, methods, attributes and the relation between the classes. The initial class diagram generated after performing NLA is shown below:



I have identified a total of 7 classes but some classes might be added or merged into one in the final class diagram. I have shown the classes, attributes, operations and relations between them.