

# PETPAL- ANIMAL ADOPTION AND RESCUE PLATFORM

---

SOFTWARE ENGINEERING PROJECT REPORT

[Submitted in partial fulfillment]

As a part of the curriculum of  
B.SC. (H) COMPUTER SCIENCE



---

Submitted by:

SANSKRITI (20075570055)

AKANKSHA ARORA (20075570004)

SANIA PAL (20075570054)

RIDDHI KASHYAP (20075570051)

**B.Sc. (H) COMPUTER SCIENCE**

**IV SEMESTER**

SHYAMA PRASAD MUKHERJEE COLLEGE FOR WOMEN, University of Delhi

# ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. We would like to show our gratitude to Mr. Lavkush Gupta for giving us a good guidance for this project throughout the numerous consultations which made us complete the project duly.

We would also like to thank our Department of Computer Science, Shyama Prasad Mukherjee College, University of Delhi , for giving us the right kind of opportunity for the successful completion of our project titled “PETPAL” and guiding us with the correct process. Many people, especially our classmates and team members itself have made valuable comments and suggestions on this proposal which gave us an inspiration to improve our work.

I must also thank my parents for the immense support and help during this project. Without their help, completing this project would have been very difficult.

Sanskriti (20075570055)

Akanksha Arora (20075570004)

Sania Pal (20075570054)

Riddhi Kashyap (20075570051)

# CERTIFICATE

This is to certify that the content of this project report entitled, “PETPAL” has been submitted by Sanskriti, Akanksha Arora, Sania Pal, Riddhi Kashyap of Shyama Prasad Mukherjee College, New Delhi for consideration in partial fulfilment of the requirement of the curriculum of B.Sc. (Hons) Computer Science.

It embodies the work done by them during semester IV of their course under the supervision of Mr. Lavkush Gupta from Shyama Prasad Mukherjee college for women.

Certified by:

Mr. Lavkush Gupta Assistant Professor,  
Department of Computer Science  
Shyama Prasad Mukherji College,  
Delhi University

## TABLE OF CONTENTS

S.No	Contents	Page No.
1	Problem Statement	06
2	Process Model	10
3	Project Scheduling	11
4	Timeline Chart	13
5	Software Requirements Specification (SRS)	15
6	Feasibility Study	23
7	Use Case Diagram	28
8	Use Case Descriptions	29
9	Context Level Diagram	35
10	Data Flow Diagram (Level-1)	36
11	Entity Relationship Diagram	37
12	Data Dictionary	38
13	Sequence Diagram	41
14	Screens/Interface Designs	42
15	Function Point Metrics	52
16	Effort Estimation using COCOMO model	56
17	Risk Analysis	59
18	Testing	62
19	Bibliography	66

# PROBLEM STATEMENT

“PetPal” is software which mainly focuses on animal adoption and animal rescue. It works on a motto “BE THEIR VOICE: THEY CAN’T SPEAK, BUT YOU CAN”. It is a place where stray, lost, abandoned, or surrendered pets are brought and served. We are also connected with Animal Welfare organizations. The software allows the users to adopt a pet, volunteer, purchase products related to pets and donate the desired amount to the organization. The software contains pets’ related products such as food, accessories, grooming products etc. to purchase provided by small scale organisations. The software also provides guidelines to the users regarding health care of pets, how the pets need to be nurtured, how they have to be taken care of after being adopted.

## ACTORS:

1. Users
2. Administrator

## FUNCTIONALITIES:

This software has the following functionalities:

- ❖ **Sign Up:** This functionality allows the user to register into system using his/her name. Also, the user needs to give their personal details, their Phone number and their E-mails for verification, along with their detailed information. This is followed by the verification, thus making the software more secure and authorized.
  - **NAME**  
The user id or the valid username of donor or receiver must be entered.
  - **PHONE NUMBER**  
The user needs to enter the valid phone number.
  - **EMAIL**  
The user needs to enter the valid email id.
  - **VERIFICATION**  
The user needs themselves to be verified via registered Phone no. through **OTP**
  - **PASSWORD FOR LOGIN**  
Users will be provided with their udername and password. They may change their passwords after first login

- ❖ **LOGIN:** This functionality helps the users to login into PetPal.
  - **Email/Phone/Username**  
User will enter their registered email or phone no. or username.
  - **Password**  
User will enter their password for successful login.
  - **Forget Password**  
If user forgets their password they can recover it.
  
- ❖ **What would you like to do?** Users can select a desirable option which suits them.
  - I want to adopt
  - I want to rescue
  - I want to donate
  - Shopping
  - Guidelines
  - My profile
  
- **I want to adopt:** User should fulfill the requirement before adopting.  
After verification user can enjoy adopting.
  1. **Location:** user will enter his location for nearby results
  2. **Select And Contact:** user will select and contact the user
  
- **I want to rescue:** User will enter his/her location then various nearby NGO's will pin down and user can contact any of them
  1. **Location**
  2. **Select nearby NGO's**
  
- **I want to donate:** User will upload photo of pet and other details
  1. **Upload:** User will upload photo of pet and write down other details
  2. **Donate money:** user will select an NGO out of the list of NGOs and can donate any amount of money.
  3. **Food and other stuff:** user can contact the NGO to donate them food or any other stuff they'd like to donate.
  
- **Shopping:** Additional option for users -> they can shop for their new pet
  1. Clothing
  2. Accessories

### 3. Food

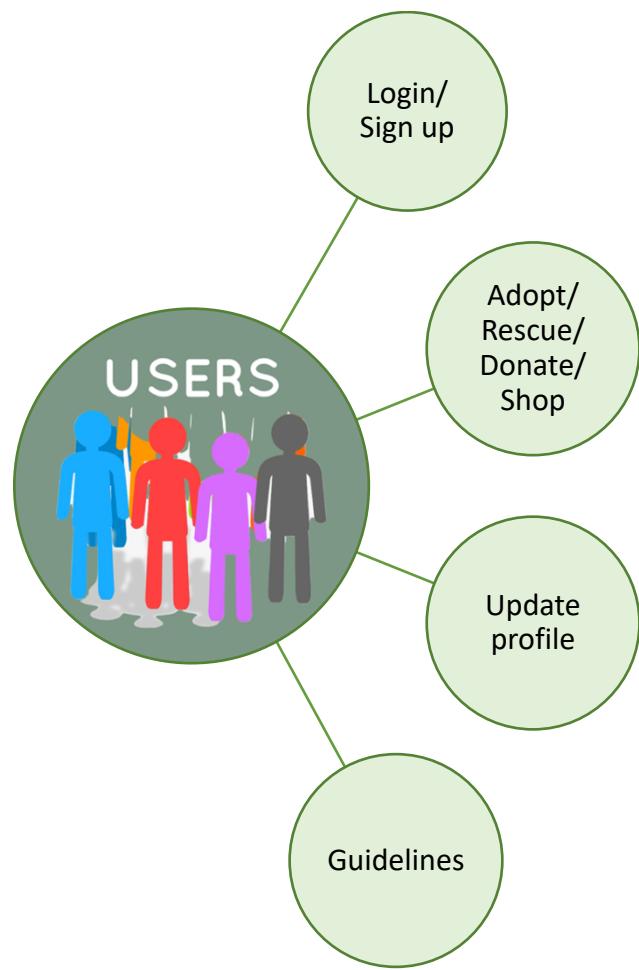
- **Guidelines:** guidelines to the user regarding health care of pets, how the pets need to be nurtured, how they have to be taken care of after being adopted.
- **My Profile:** This functionality would help users update their profile provide the knowledge about the software and its functionalities.
  1. **Update Profile**
  2. **Change Phone no.**
  3. **Change Password**
- **Help and support:** The user may get the help about the software through this option
  1. Community guidelines
  2. FAQ
  3. Helpline number

## ACTORS:

1. **ADMINISTRATOR:** The administrator is the super user of this application. The admin has all the information about all the users and about all products. It manages the database and all other data related actions of the users.



2. **USERS:** The end users are the ones using the application, who visit the websites and to whom the services are being provided.



# PROCESS MODEL

PETPAL System follows Incremental Process Model.

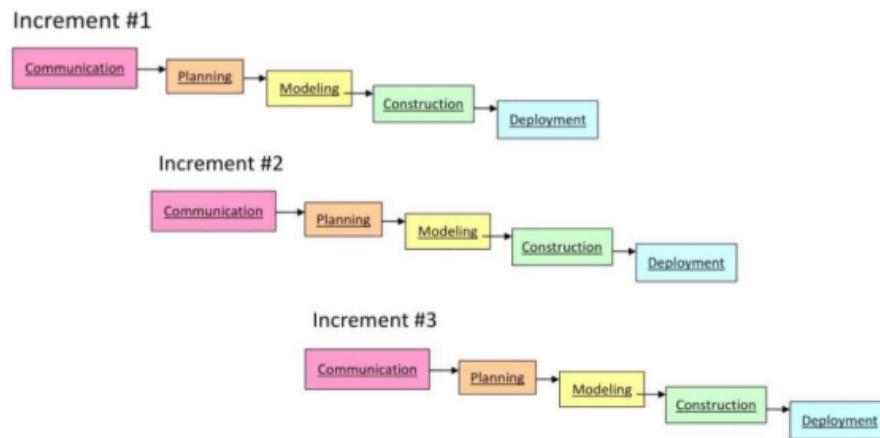
Incremental process model is also known as the Successive version model.

We have used the incremental model as it combines elements of linear and parallel process flows. It generates working software quickly and early during the software lifecycle. This model is more flexible and less costly to change scope and requirements. It is easier to test and debug during a smaller iteration.

First, a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations/ versions are implemented and delivered to the customer until the desired system is released.

In this model, the customers can respond to each built. Also, functionality can be refined and expanded in the later stages of the later software releases. The user can visualize the software before the completion of the entire project in order to evaluate and provide feedback. We are using this model as requirements are completely understood, however, small changes can be incorporated.

## Incremental Model (Diagram)



# PROJECT SCHEDULING

Work tasks	Planned start	Actual start	Planned complete	Actual complete	Assigned person	Effort associated
Meeting with customers milestone: problem statement	Jan wk 1	Jan wk 1	Jan wk 2	Jan wk 2	Sanskriti Akanksha Sania Riddhi	4 p-w
Software process model	Jan wk 2	Jan wk 2	Jan wk 3	Jan wk 3	Sanskriti Akanksha Riddhi	3 p-w
Project scheduling	Jan wk 3	Jan wk 4	Jan wk 4	Jan wk 4	Sania Riddhi	2 p-w
Identifying needs validation milestone: SRS documented	Jan wk 4	Jan wk 4	Feb wk 1	Feb wk 2	Sanskriti Sania	2 p-w
Screens/ Interface Designs	Feb wk 1	Feb wk 1	Feb wk 2	Feb wk 2	Sanskriti Akanksha Sania Riddhi	4 p-w
ERD	Feb wk 2	Feb wk 2	Feb wk 3	Feb wk 3	Sanskriti Akanksha	2 p-w
Data dictionary	Feb wk 2	Feb wk 2	Feb wk 3	Feb wk 3	Akanksha	1 p-w
Data flow diagram (level zero) (Context level diagram)	Feb wk 4	Feb wk 4	March wk 1	March wk 1	Sania Riddhi	2 p-w
Data flow diagram (level 1)	March wk 1	March wk 1	March wk 2	March wk 2	Sanskriti Akanksha	4 p-w

					Sania Riddhi	
Use diagram	March wk 2	March wk 2	March wk 3	March wk 3	Sania Riddhi	2 p-w
Use case description	March wk 2	March wk 2	March wk 4	March wk 4	Akanksha Riddhi	2 p-w
Sequence diagram	April wk 1	April wk 2	April wk 2	April wk 2	Sanskriti Sania	2 p-w
Function point metric	April wk 2	April wk 3	April wk 3	April wk 3	Sanskriti Akanksha Riddhi	3 p-w
COCOMO model	April wk 2	April wk 3	April wk 3	April wk 3	Sanskriti	1 p-w
Risk analysis	April wk 2	April wk 3	April wk 3	April wk 3	Akanksha Sania	2 p-w
Testing	April wk 3	April wk 4	April wk 4	April wk 4	Sanskriti Sania Riddhi	3 p-w

# TIMELINE CHART

WORK TASKS	JANUARY				FEBRUARY				MARCH				APRIL			
	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4
Problem Statement																
Software Lifecycle Model																
Project Scheduling																
SRS																
SCREENS																
ERD																
DATA DICTIONARY																
Context Level Diagram																
DFD 1																
Use Case Diagram																
Use Case Description																
SEQUENCE DIAGRAM																
Project Metrics																

<b>Effort estimation (COCOMO)</b>															
<b>Risk Analysis</b>															
<b>Testing</b>															

# **SOFTWARE REQUIREMENTS SPECIFICATION (SRS)**

## **1. INTRODUCTION:**

This section gives a scope description and overview of everything included in the SRS document. Also, the purpose of this document is described.

### **1.1 Purpose**

The purpose of this document is to provide a detailed description of the requirements of "PetPal" software.

This software mainly focuses on animal adoption and animal rescue. It works on a motto “BE THEIR VOICE: THEY CAN’T SPEAK, BUT YOU CAN”. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interfaces and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team. Both the stakeholders and the developers of the system can be benefited from this document.

### **1.2 Scope**

The main objective of this project is to automate the process of serving towards the welfare of the pets by giving the abandoned pets a place of shelter, care for them with affection. The scope of the project is to provide ease of working towards the welfare of abandoned, stray, lost, or surrendered pets. It saves time, paperwork is less, and easy retrieval of records, automating the system reduces the process of middle man who takes a commission.

### **1.3 References**

[1] IEEE Software Engineering Standard Committee”, IEEE Std. Recommended Practice for Software Requirements Specification”, 2011

### **1.4 Overview**

The rest of the SRS document describes various system requirement, interfaces, features and functionalities in detail.

## **2. THE OVERALL DESCRIPTION**

This section will give the overview of the whole system. It basically manages the functionality of the system in terms of hardware, software, human ware. It is a pet-friendly application where the users can get connected, share their pet's pictures, and also share the picture of some abandoned animals they come across, through which a shelter can be searched for the poor animal.

### **2.1 Product perspective**

PetPal is the window based, self-contained and dependent system. This system involves three end users i.e. rescuer, adopter and NGO.

#### **2.1.1 User Interface:**

It should be the bridge between the various parts of the system, between other parts or unit of the system. The user interface must be implemented using any tool or software package like Java Applet, MS Front .

#### **2.1.2 Hardware Interface:**

Since the application will run over the internet, all the system using this application may need to connect with internet and this will be hardware interface as for example Modem, WAN -LAN, Ethernet, Cross-C.

## **2.2 Product Functions:**

Users must provide correct details, a valid e-mail or a phone number to receive OTP for registering into the application. When an invalid OTP is entered or any other issue with OTP verification occurs, the user gets a pop-up message with an option for re-sending the OTP. With this the user will get their user ids and passwords so as to login into the system. All the users will be authenticated to avail the service.

## **2.3 Actors**

There are 2 types of actors in this software:

- User: using the software to adopt/rescue strays.
- Admin: can add/delete/edit/maintain records.

## **2.4 Constraints**

- The interface will be in English only.
- This supports only android and web.
- Crucial to have an internet connection.

## **2.5 Assumptions and Dependencies**

The product does require back-end database server MySQL for storing the username and password and valid email id for different types of users of the system.

### **2.5.1 Assumptions:**

- User must be trained for basic computer or mobile functionality.
- User must have basic knowledge of English.

## **3 SPECIFIC REQUIREMENTS**

### **3.1 External Interfaces**

The application will have menu based interface. There are many types of interfaces which are such supported by the PetPal software system.

#### **3.1.1 Hardware Interface:**

Since, the application works online i.e. runs over internet, all the hardware shall require to connect internet will be hardware interface for the system. For Ex- Modem, WAN -LAN, Ethernet, Cross-Cable.

#### **3.1.2 Software Interface:**

This includes the embedded application that will be used in supporting the various functions of the system.

- A GPS for detecting the location or the direction.
- Operating system – Windows
- Google maps for tracking the location.
- Visual Basic 6.0(for developing the software)
- A web server and Database at the server side and Telecommunication like a service provider

### **3.2 Functions:**

The software provides the following functionalities: -

❖ **Sign Up:** This functionality allows the user to register into system using his/her name. Also, the user needs to give their personal details, their Phone number and their E-mails for verification, along with their detailed information. This is followed by the verification, thus making the software more secure and authorized.

- **NAME**

The user id or the valid username of donor or receiver must be entered.

- **PHONE NUMBER**

The user needs to enter the valid phone number.

- **EMAIL**

The user needs to enter the valid email id.

- **VERIFICATION**

The user needs themselves to be verified via registered Phone no. through **OTP**

- **PASSWORD FOR LOGIN**

Users will be provided with their id and password. They may change their passwords after first login

❖ **LOGIN:** This functionality helps the users to login into PetPal.

- **Email/Phone**

User will enter their registered email or phone no.

- **Password**

User will enter their password for successful login.

- **Forget Password**

If user forgot their password they can recover it.

❖ **What would you like to do?:** Users can select a desirable option which suits them.

- I want to adopt
- I want to rescue
- I want to donate
- Shopping

- Guidelines
- My profile

 **I want to adopt:** User should fulfill the requirement before adopting. After verification user can enjoy adopting

- **Location:** user will enter his location for nearby results
- **Select And Contact:** user will select and contact the user

 **I want to rescue:** User will enter his/her location then various nearby NGO's will pin down and user can contact any of them

- **Location**
- **Select nearby NGO's**

 **I want to donate:** User will upload photo of pet and other details

- **Upload:** User will upload photo of pet and write down other details

 **Shopping:** Additional option for users -> they can shop for their new pet

 **Guidelines:** guidelines to the user regarding health care of pets, how the pets need to be nurtured, how they have to be taken care of after being adopted.

 **My Profile:** This functionality would help users update their profile provide the knowledge about the software and its functionalities.

- **Upload Profile**
- **Change Phone no.**

- **Change Password**

 **Help and support:** The user may get the help about the software through this option

### **3.3 Performance Requirements**

The product shall be based on web and has to run from a web server. The product shall take initial load time depending on internet connection strength which also depends on the media from which the product is run. The performance shall depend upon hardware components of the device used by the user.

### **3.4 Design Constraints**

- This software can be installed on Personal Computers, tablets, or smart phones.
- For security reasons, login id and password must be provided.

### **3.5 Software System Attributes**

#### **3.5.1 Security:**

The application will be protected by a username and password.

#### **3.5.2 Portability:**

The application is portable.

#### **3.5.3 Reliability:**

The application is reliable as any failure in it can be repaired in 2-3 days.

### **3.5.4 Availability:**

This application's server is working the whole 24x7 time.

### **3.6 Other Requirements:**

Performance, throughput, scalability.

## **4. CHANGE MANAGEMENT PROCESS:**

The system must allow the additional features to be integrated in the near future.

# FEASIBILITY STUDY

A feasibility study assesses the technical, operational, and economic merits of the proposed project. It helps in determining whether a project will be effectively possible or not.

## **1. Technical feasibility:**

The need for a PetPal application is that it allows animals and users to get matched to one another based on their characteristics and needs including lifestyle and personality. It does lead to several challenges both in the interface and the backend of the system.

The following are the technical challenges identified for the PetPal below:

- The PetPal will need a secure interaction and user authentication on the web application.
- The PetPal will need a method to communicate with a backend database.
- The PetPal will need a way for retrieving the information from a database and displaying it in a clean and organized layout.
- The PetPal will need a way to provide a responsive interface onto a wide variety of platforms (personal computer, tablet, smartphone) that allows users to create a profile including information on lifestyle and personality.
- The PetPal will need a way to find potential animal matches within the system.
- The PetPal will need to match users to animals based on profiles.

→ The PetPal will need a way to ingest large database files such as CSV (comma separated value), excel sheets.

## 1. Hardware Requirements

- Processor: Intel(R) Pentium(R) or above
- CPU Hard disk: 500GB HDD
- RAM: 4GB RAM

## 2. Software Requirements

- Operating System: Windows 7 and above
- Front End: HTML, Python
- Back End: python GUI, tkinter
- Database : MySQL, Django, divio

Each of the technologies are freely available and the technical skills required are manageable.

METHODS	PROS	CONS
HTML	<ul style="list-style-type: none"><li>▪ Portable</li><li>▪ Free</li><li>▪ Every browser supports HTML language</li><li>▪ HTML allows utilization of templates which makes webpage designing easy.</li></ul>	<ul style="list-style-type: none"><li>▪ Need to write large codes for making a simple webpage</li><li>▪ Security features are good in HTML</li></ul>
Python	<ul style="list-style-type: none"><li>▪ Free and open source</li></ul>	<ul style="list-style-type: none"><li>▪ Slow speed</li></ul>

	<ul style="list-style-type: none"> <li>▪ Vast library support</li> <li>▪ Portable</li> </ul>	<ul style="list-style-type: none"> <li>▪ Weak in mobile computing</li> <li>▪ Not memory efficient</li> </ul>
MySQL	<ul style="list-style-type: none"> <li>▪ Portability</li> <li>▪ Most secure and reliable database management system</li> </ul>	<ul style="list-style-type: none"> <li>▪ Does not have good developing and debugging tool compared to other databases.</li> </ul>

## SUMMARY

ISSUE	SOLUTION
We need a web frame that will allow us to easily build and create the website.	Django, easy to understand python-based web framework
The content management system that we use must be easily maintainable for our client.	Divio, an easy-to-use CMS that is compatible with Django.
The database must be flexible and easy to build upon.	MongoDB, a flexible, non-relational database.
We must be able to connect our chosen database to our web framework.	We will use a third-party engine MongoDB-engine, to integrate MongoDB with Django.

We are confident that we will implement this project given the technologies described above. One uncertainty that still remains is how the current animal data is being kept. However, we have prepared ourselves with a few alternative solutions for moving

over that data, so whatever the format may be, it should not be a problem.

### **3. Operational Feasibility:**

PetPal is an application to adopt/rescue strays, and such an application is very much necessary in today's times, as people are now more aware and concerned towards the strays but don't have enough resources to help them. With this application, we are providing them the resources and making it easier for them to contact NGOs to rescue/adopt/foster strays.

- It is more convenient and time saving than searching contact information of NGOs online and contacting them personally one by one manually.
- There are many instances where people find strays and want to help them but are not quite sure how to, with this application they will be able to contact NGOs with just a click for them to foster the strays. This saves them a lot of time.
- It is a one stop destination for every pet lover/owner; pets, pet supplies, after-care of pets everything is available at one place, every step is done online at the user's convenience.
- A fully secured system enables the users to connect with us without letting them worry about frauds or any other misconduct.

### **4. Economic Feasibility:**

- It refers to the analysis of the cost-effectiveness of a project.
- PetPal is economically feasible as it uses free and open-source technologies.

- It is an interactive application with free sign-up.
- The system will follow the freeware software standards, no cost will be charged from the potential customers.
- One time investment to the required software and hardware for our developers will be made.

## **5. Legal Feasibility:**

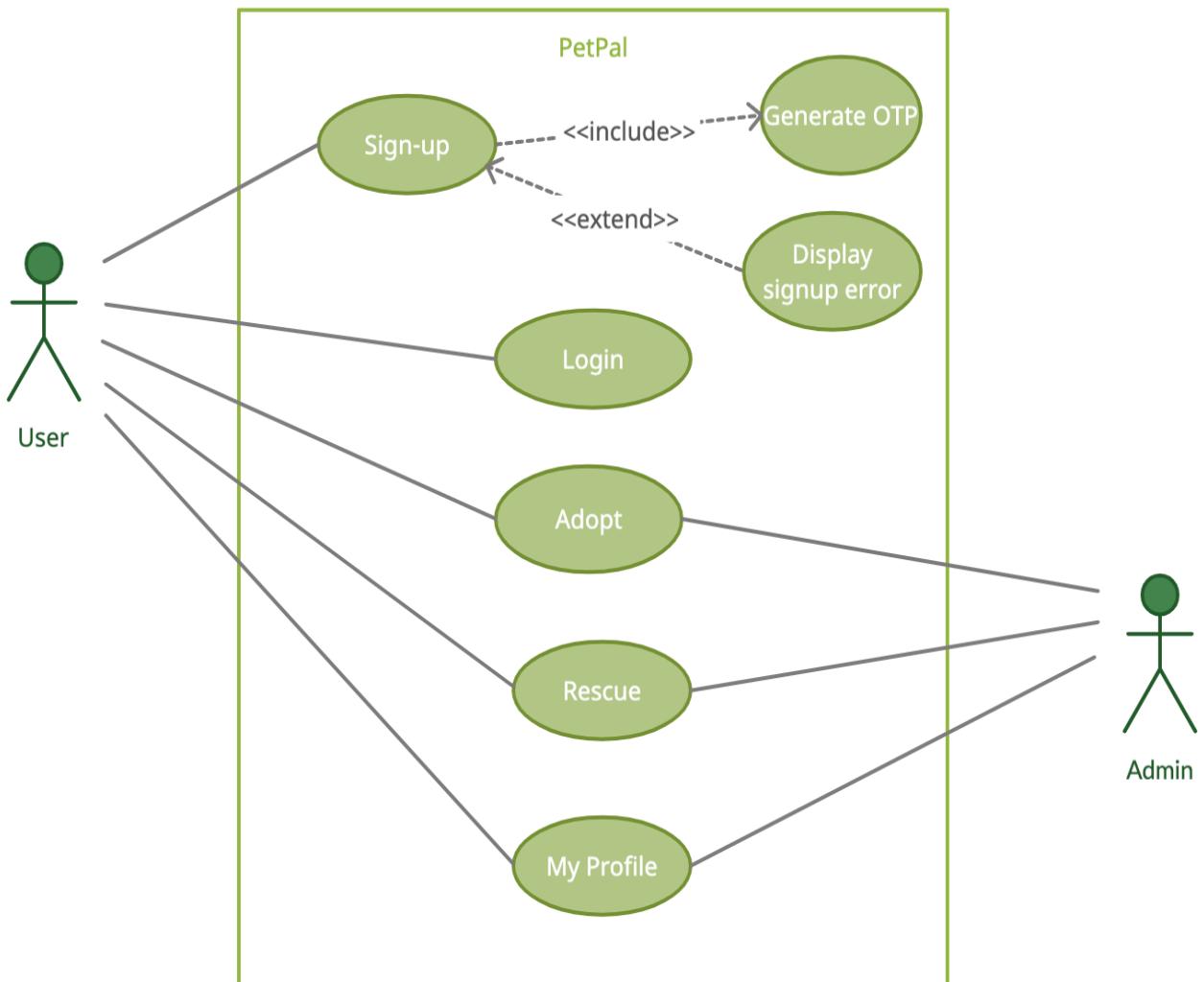
- PetPal uses free development tools and only the programmers will have access and can modify the software.
- Software libraries that are used in this system are free open source libraries.

## **6. Security Feasibility:**

PetPal is an application to adopt rescues strays which includes the interaction of users with one another with which the responsibility of creating a secure environment is necessary, therefore:

- Filtering out and reporting any post will be available with an immediate action taken against them.
- Not following the basic protocols will not be entertained.
- Repeated reports against any user will result into deactivating their account after giving a warning.

# USE CASE DIAGRAM



# USE CASE DESCRIPTIONS

## USE-CASE (BRIEF)

- Sign up:  
Sign up provides new user to first register in the software via mail/OTP generation
- Login system:  
Login is a way to enter through user id and password/ Forgot and reset password feature . The function allows the user to register as a member to interact with the system  
Error: If user enters wrong id or password, display login error
- Rescue:  
Designed to let people upload/post the animal's profile which includes all the details of that animal like age, breed, colour etc and let the adoption party knows about the animal for clear and better understanding.
- Adopt:  
Let people adopt the animal they feel connection with and can take the best care of.

## USE-CASE DESCRIPTIONS

### 1. Signup system:

1.1 Introduction: Allows the new user to first register in the software via mail. Here users can create their profile by specifying the unique username, phone number and password.

1.2 ACTORS: User/ Admin.

1.3 Pre-Conditions: None

**1.4 Post-Conditions:** If the use case is successful, the actor can successfully login anywhere and anytime and can utilize the facilities provided by PetPal.

**1.5 Basic Flow:**

(i) System needs the actor to enter his/her name, user name, email id, generate his/her password, phone number and their location.

**1.6 Alternate Flow:**

(i) Compulsory options left, the system displays ERROR. The actor then have to again fill that particular option.

(ii) Invalid Phone number/ email id/ password, the system displays ERROR. The actor then again have to fill that particular option

## **2. Login system:**

**2.1 Introduction:** Describes how user can create their profile in PetPal.

**2.2 ACTORS:** User/ Admin

**2.3 Pre-Conditions:** User must be registered in petpal.

**2.4 Post-Conditions:** If the use case is successful, the actor is logged into the system. If not, the system state is unchanged.

**2.5 Basic Flow:**

(i) System requests that the actor enter his/her username and password.

(ii) The actor enters his/her name & password.

(iii) System validates name & password, and if finds correct allow the actor to logs into the system.

**2.6 Alternate Flow:**

(i) Invalid user name, the system displays ERROR. The actor can choose to either return to the beginning of the basic flow or cancel the login, at that point, the use case ends.

(ii) Invalid password, the system displays ERROR. The actor can choose to either return to the beginning of the basic flow or can choose the forgot password option to reset the password and then actor gets verification mail/ OTP in his/her number or email id.

(iii) Not registered, the system displays ERROR. The actor can choose to either return to the beginning of the basic flow or cancel the login, at that point, the use case ends.

### 3. Rescue:

3.1 Introduction: This functionality is used to let people post or upload the animal's profile who needs their \*forever\* home

**Animal's profile:** to upload about an animal who is up for adoption, animal's id will be made which will include all the information about the animal, it's name, breed, gender, age etc.

**Edit profile:** actor can edit profile later on if they want to modify the description and can also delete the profile as they required.

**Reply to adopter:** If any user requests the rescuer to adopt their respective animal, they can reply to their request.

#### 3.2 ACTORS: User

##### 3.3 Pre-Conditions:

- (i) User must be registered in petpal.
- (ii) Petpal uses location to help other adopters filter the search results to their requirements therefore location must be added.

3.4 Post-Conditions: if use case is successful, a post of the animal up for adoption will be uploaded and be shown in the feed.

##### 3.5 Basic Flow:

(i)Animal's profile: user can create the profile of their animal which is up for adoption by adding the basic details and the description necessary for others to adopt.

(ii)Edit profile: actor has flexible options to edit their animal's profile according to their need.

(iii)Reply to adopter: when any user requests the rescuer for adoption for their pet, the rescuer can see the adopter's profile and manage the situation and take the decision as per their satisfaction and requirements.

##### 3.6 Alternate Flow:

(i) basic details of the animal is necessary, if not given the animal's profile will not be created.

(ii)if the rescuer due to some conditions don't want to display their animal's profile on the feed but forgets to delete the created profile, they would still be receiving the requests from the potential adopters.

#### 4. Adoption:

4.1 Introduction: This functionality allows user to adopt the animal they'd like to give a home to. The user can register themselves into the system to view the list of animals waiting to be adopted along with each animal's record such as their age, gender, description.

**Feed:** user will be shown the feed of the pets profile waiting to be adopted

**Search key:** apart from the feed, user can also search the type or breed of the animal they'd like to adopt

**Filter key:** this feature helps user to meet their specifications

**Send request:** if user finds the animal they'd like to adopt they can send the request to the user with which the animal belongs to

**Status:** depends on the rescuer if they'd like to process further with the requesting user,

#### 4.2 ACTORS: User

#### 4.3 Pre-Conditions:

(i) User must be registered in petpal.

4.4 Post-Conditions: Either gets the pet or not.

#### 4.5 Basic Flow:

(i)Feed: a feed of animal's profile uploaded by the rescuer will be shown after the homepage, those animals will be displayed with their details to make it easy for adopters.

(ii)search key: user can search the type or breed or any other main detail they definitely want in their pet to adopt. They will be shown all the animals satisfying the filter specifications.

(iii)filter key: other than the search key, user will have the option to filter out minor details they want such as age color etc,

(iv) send request: if the user wants to adopt an animal they can send request to the user it belongs to, they will receive the notification and then they can further contact for the same.

(v) status: accepted/ declined/ in process are the three status user will be shown after they have sent the request. It will help them to understand at what prospect they stand. Accepted will be shown when their request will be approved by the rescuer. Declined will be shown if the request will be rejected. And in-process will be shown if no action has taken against/for the request.

#### 4.6 Alternate Flow:

- (i) rescuer decline the request of adopter.
- (ii) adopter may want to backoff their request.

### 5. My Profile

**5.1 Introduction:** This functionality allows user to have a brief about their activity on Petpal. It will allow user to check which animal and how many requests they have already sent to adopt the animal, or as a rescuer they will see how much notifications they have on their respective pet.

**History:** user will be shown the number of requests they have sent and received on the respective animal.

**My profile:** user will be shown their name and locations with which they are active on petpal. The same name will be shown to other users. Apart from name, user's mobile number will be shown to the respective person only to make sure with which number they have logged in.

**Edit my profile:** this feature helps user to edit their profile if they want to specify some changes.

#### 5.2 ACTORS: User

#### 5.3 Pre-Conditions:

- (i) User must be registered in petpal.

**5.4 Post-Conditions:** gets to know their activity on Petpal

#### 5.5 Basic Flow:

(i)History: user can see the stage of their application and If they are rejected or accepted, and the rescuer can check their pet's application and the number of requests.

(ii)my profile: user can see their profile which includes their name and phone number .

(iii)edit my profile: user can edit if the name and phone number is not what they wanted to show.

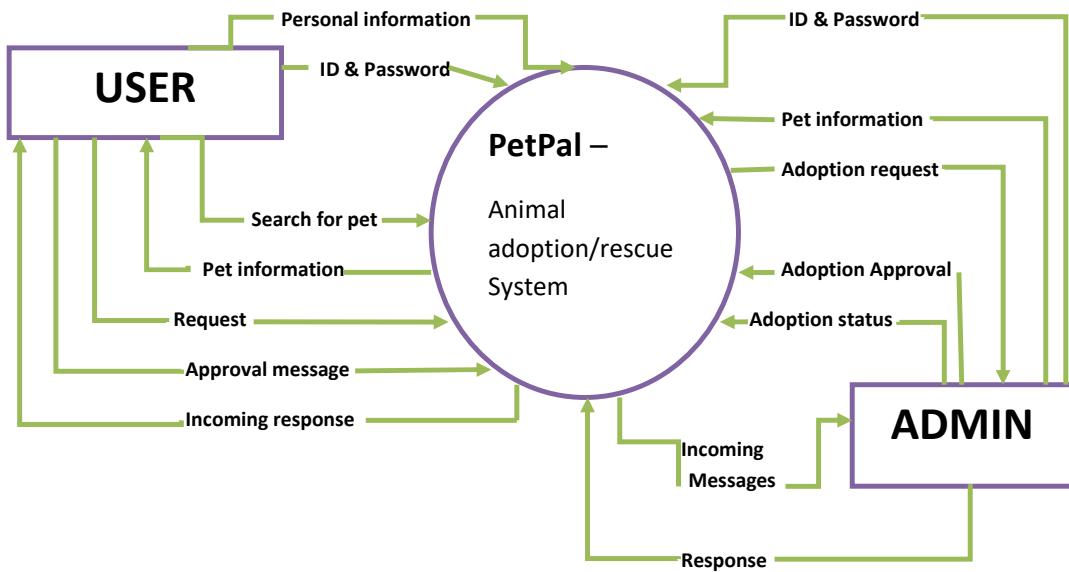
#### 5.6 Alternate Flow:

(i) user puts the wrong name in the first place.

(ii)user puts out the wrong information of their pet.

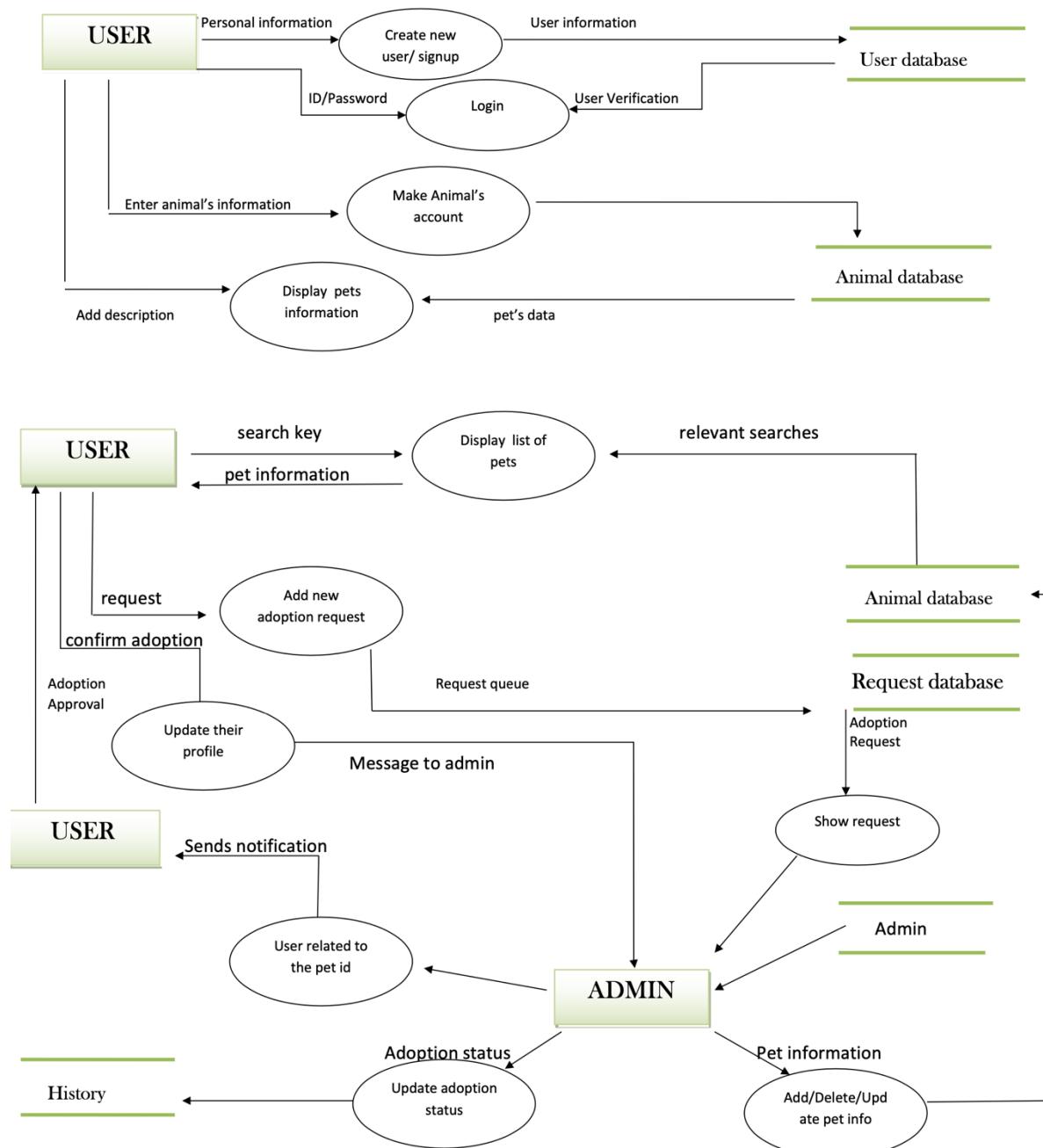
# CONTEXT LEVEL DIAGRAM

## CONTEXT LEVEL DIAGRAM(0-Level)



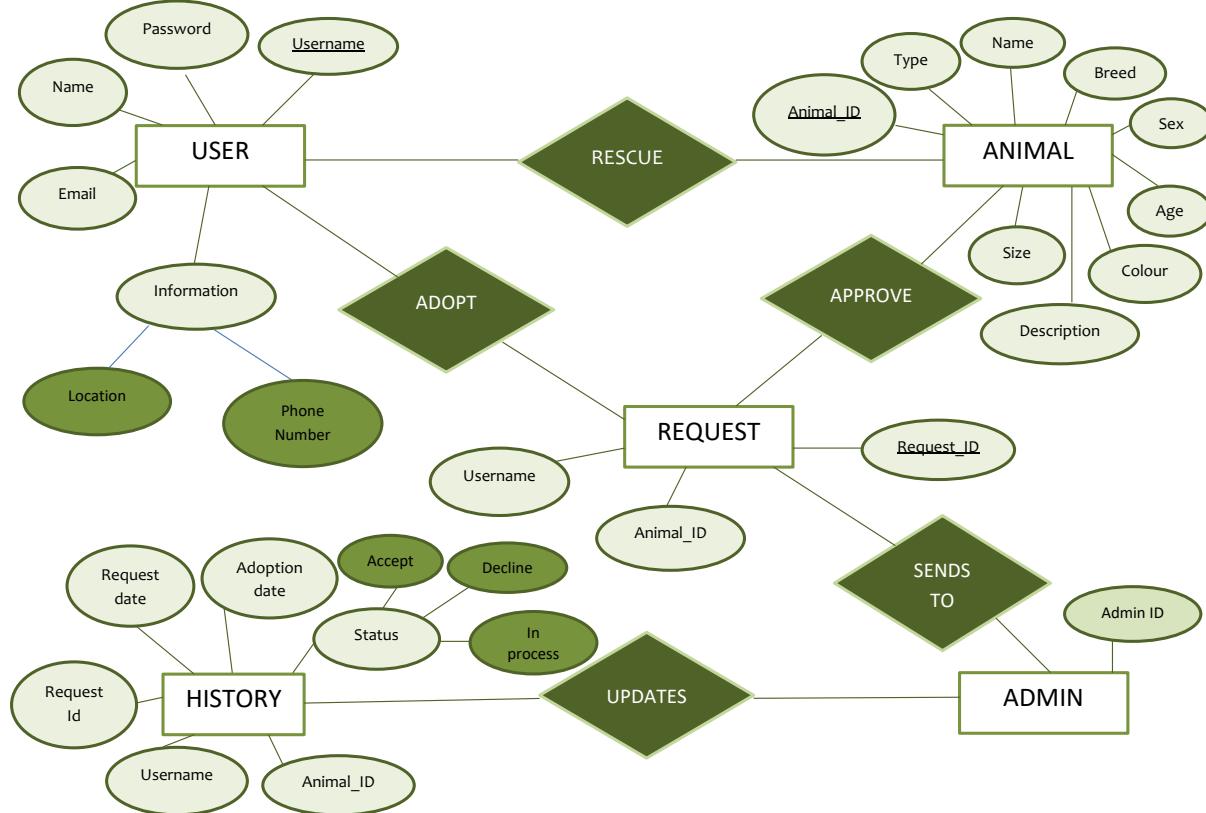
# DATA FLOW DIAGRAM (LEVEL-1)

**DFD(Level-1)**



# ENTITY RELATIONSHIP DIAGRAM

ER DIAGRAM



	Degree
RESCUE	2
ADOPT	2
APPROVE	2
SENDS	2
UPDATES	2

# DATA DICTIONARY

## USER

S. No	FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTON	EXAMPLE
1.	NAME	varchar	40	Name of the user	Sania, Riddhi
2..	EMAIL	varchar	50	Registered email-id they prefer	aka02@gmail.com
3.	USER ID (PK)	int	6	Unique ID given to user	1001, 1690
4.	PASSWORD	varchar	20	A key to login to the user's account	XYZ10, @SANIA01
5.	GENDER	varchar	10	How to identify user	M, F, NB, Rather not say
6.	LOCATION	varchar	80	Their address	1-89/new delhi
7.	PHONE NUMBER	char	12	Mobile number they prefer	96582---

## ANIMAL:

S.NO.	FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTION	EXAMPLE
1.	ANIMAL_ID (PK)	varchar	4	Unique key to identify each ID	D16, C32, A49
2.	NAME	varchar	20	Name of the pet	Bruno, noodle
3.	TYPE	char	20	Type of the animal	Dog, Cat, hamster, any other

4.	BREED	char	20	Breed of the pet	Labrador, Persian cat etc
5.	SEX	char	1	Gender of the pet	M,F
6.	AGE	char	2	Age specification	2 year, 8 months
7.	COLOUR	char	10	Colour of the pet	Black, White
8.	SIZE	varchar	10	Size of the animal	Large breed, 20 cm
9.	DESCRIPTION	Varchar	80	Qualities of the pet	Friendly, energetic etc

#### REQUEST:

S.NO.	FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTION	EXAMPLE
1.	REQUEST_ID(PK)	varchar	10	Unique code given to user's request	1001_26, 1690_45
2.	USER.USER_ID(FK)	int	6	Unique key to identify each user	1001, 1690
3.	ANIMAL.ANIMAL_ID (FK)	varchar	4	Unique key to identify each animal	D16, C32, A49

#### ADMIN:

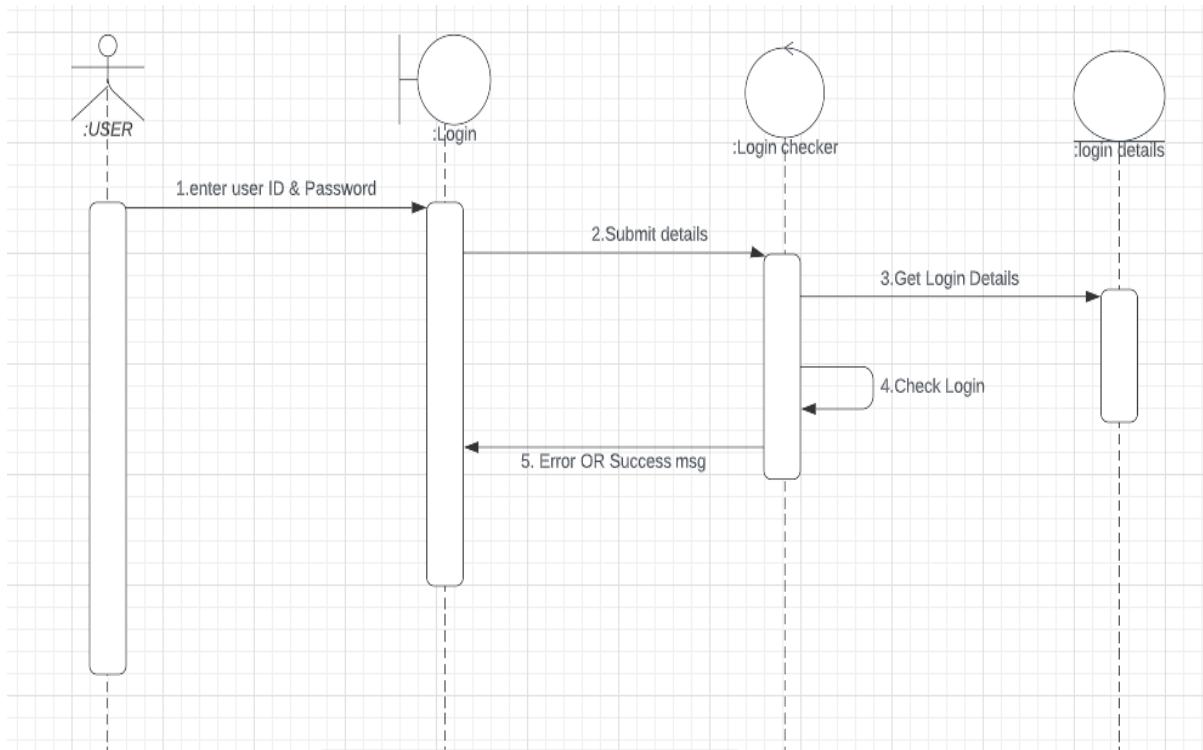
S.NO.	FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTION	EXAMPLE
1.	ADMIN_ID	varchar	40	Admin's id for login	Aka01
2.	ADMIN_PASSWORD	varchar	20	Password for login	@Aka01

HISTORY:

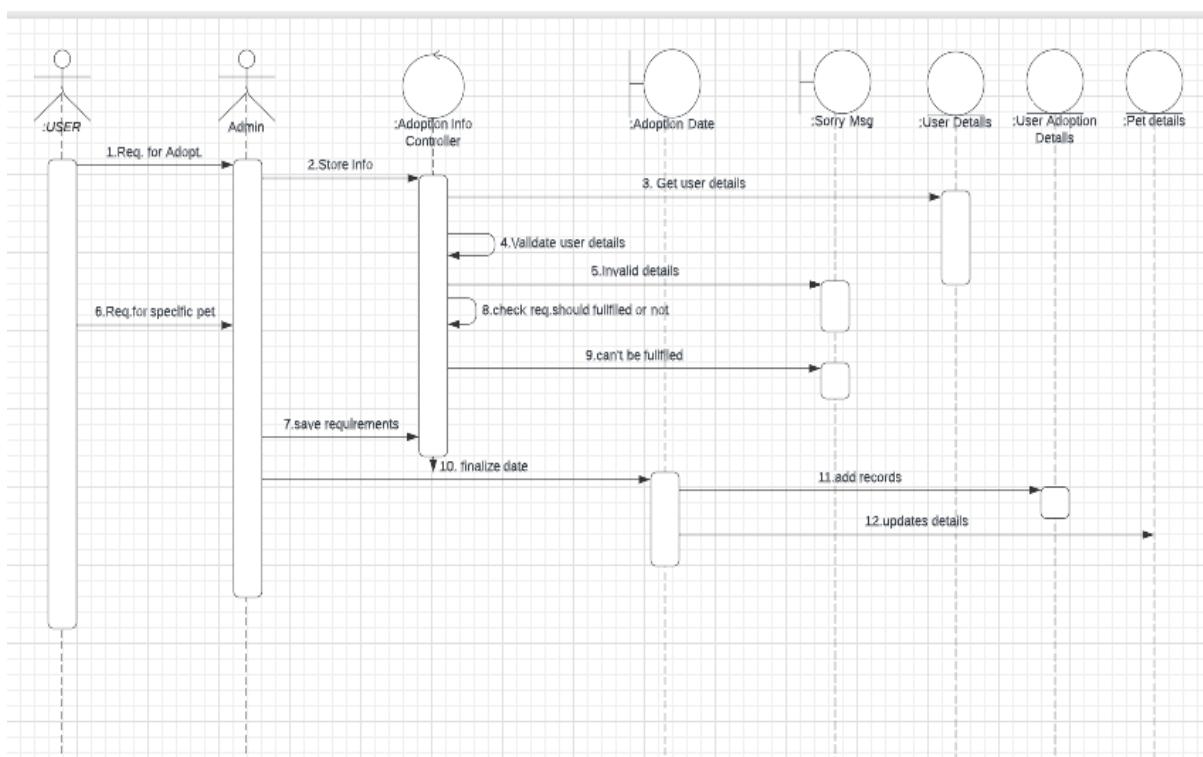
S.NO.	FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTION	EXAMPLE
1.	REQUEST DATE	char	10	History of the date request sent	2022-02-27
2.	ADOPTION DATE	char	10	History of the date request accept	2022-03-07
3.	STATUS	varchar	10	Status of the in-process	Accept, decline, in-process

# SEQUENCE DIAGRAM

## LOGIN



## ADOPTION



# SCREENS (DESIGN INTERFACES)

## 1. START UP:



No of External Inputs	0
Number of External Outputs	0
Number of External Enquiries	0
Number of Internal Logical Files	0
Number of External interfaces	0

## 2. SIGNUP



your name

username

password

re-password

email

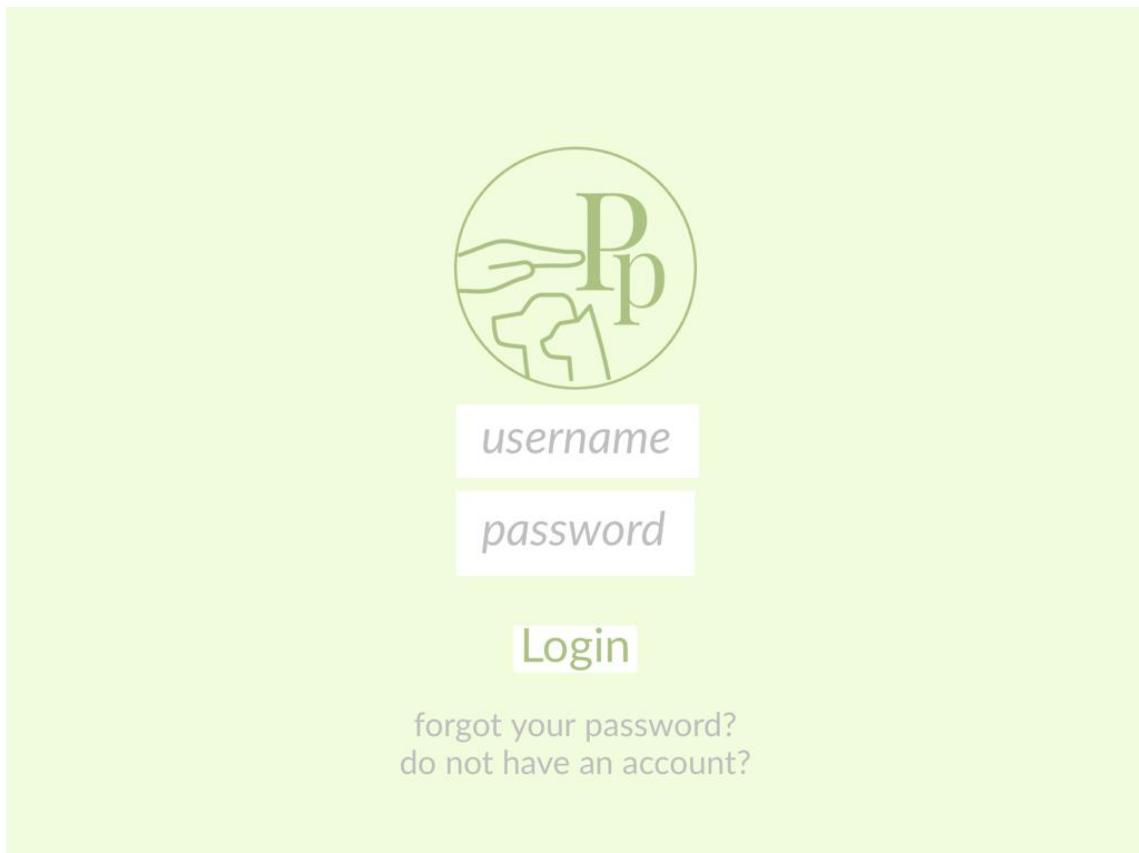
**sign up**

already a member?  
[click here to login](#)

by signing up you agree to  
our terms and conditions

No of External Inputs	5
Number of External Outputs	1
Number of External Enquiries	0
Number of Internal Logical Files	1
Number of External interfaces	0

### 3. LOGIN



No of External Inputs	2
Number of External Outputs	1
Number of External Enquiries	0
Number of Internal Logical Files	1
Number of External interfaces	0

## 4. FUNCTIONALITIES PAGE



No of External Inputs	1
Number of External Outputs	1
Number of External Enquiries	0
Number of Internal Logical Files	0
Number of External interfaces	0

## 5. SEARCH

We search hundreds  
of shelters nationwide  
to find the pet  
that's right for you

Dog 

All breeds 

Location

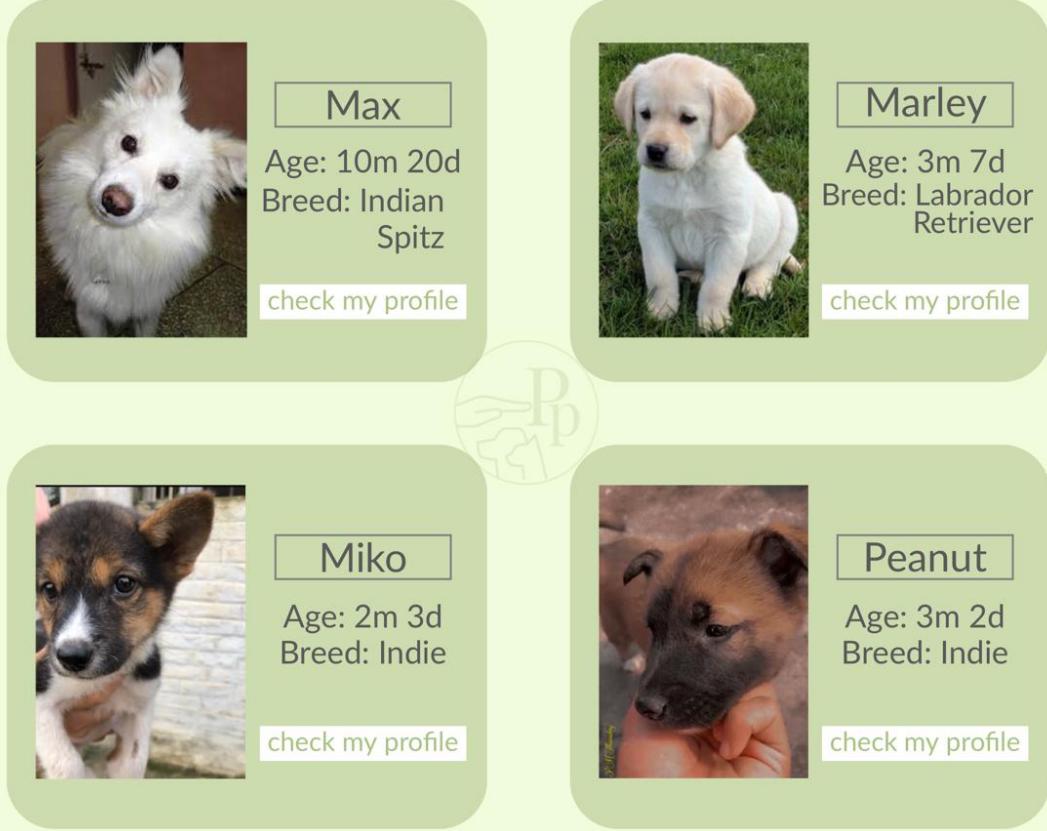
search



Find the YIN  
to your YANG  
with PetPal !

No of External Inputs	3
Number of External Outputs	1
Number of External Enquiries	1
Number of Internal Logical Files	0
Number of External interfaces	0

## 6. PETS FEED



A central circular icon containing a stylized letter 'P' with a small dog silhouette inside it.



**Max**  
Age: 10m 20d  
Breed: Indian Spitz  
[check my profile](#)



**Marley**  
Age: 3m 7d  
Breed: Labrador Retriever  
[check my profile](#)



**Miko**  
Age: 2m 3d  
Breed: Indie  
[check my profile](#)



**Peanut**  
Age: 3m 2d  
Breed: Indie  
[check my profile](#)

No of External Inputs	1
Number of External Outputs	1
Number of External Enquiries	0
Number of Internal Logical Files	1
Number of External interfaces	0

## 7. UP FOR ADOPTION?



Hello PetPal,  
My name is

**Max**

I'm 10 months and 20 days old.  
I was rescued by the team of  
PetPal in Jan 2022. They found  
me on the streets of Nehru Place,  
Delhi. Since then they've been  
taking care of me. I'm an Indian  
Spitz, so I'm bright, active and  
playful.

Would you like to  
**ADOPT ME?**

No of External Inputs	1
Number of External Outputs	0
Number of External Enquiries	0
Number of Internal Logical Files	0
Number of External interfaces	0

## 8. MY PROFILE

### My history C

 Max  
Age: 10m 20d  
Breed: Indian Spitz  
[check my profile](#)

 Mimi  
Age: 1m 3d  
Female domestic Shorthair/mix  
[check my profile](#)





## MY PROFILE

Name  
**Benny Dayal**

Location  
**New Delhi**

**EDIT MY PROFILE**

No of External Inputs	1
Number of External Outputs	2
Number of External Enquiries	0
Number of Internal Logical Files	1
Number of External interfaces	0

## 9. ABOUT US

### ABOUT US



Since 2022, helping strays to find their  
FOREVER homes.

PetPal is software which focuses on stray  
animal adoption and rescue. It works on  
a motto "BE THEIR VOICE: THEY CAN'T  
SPEAK, BUT YOU CAN".

for further queries you can

**CONTACT US!**

No of External Inputs	1
Number of External Outputs	0
Number of External Enquiries	0
Number of Internal Logical Files	0
Number of External interfaces	0

## 10. CONTACT US:



Contact us!

-  petpalorg@gmail.com
-  petpalindia
-  heypetpal
-  9876543210
-  Delhi, India

No of External Inputs	0
Number of External Outputs	0
Number of External Enquiries	0
Number of Internal Logical Files	0
Number of External interfaces	0

# FUNCTION POINT METRICS

## Estimation technique using function-point metrics

The function point (FP) metric can be used effectively as a means for measuring the functionality delivered by a system. Using historical data, the FP metric can be used to:

1. Estimate the cost or effort required to design, code, and test the software.
2. Predict the number of errors that will be encountered during testing.
3. Forecast the number of components and/or the number of projected source lines in the implemented system.

Function points are derived using an empirical relationship based on countable (direct) measures of software's information domain and qualitative assessments of software complexity.

Information domain values are defined as:

- I. Number of external inputs (EIs)
- II. Number of external outputs (EOs)
- III. Number of external inquiries (EQs)
- IV. Number of internal logical files (ILFs)
- V. Number of external interface files (EIFs)

The  $F_i$  ( $i = 1$  to  $14$ ) are value adjustment factors (VAF) based on responses to the following questions:

1. Does the system require reliable backup and recovery?

Ans. 5

2. Are specialized data communications required to transfer information to or from the application?

Ans. 3

3. Are there distributed processing functions?

Ans. 3

4. Is performance critical?

Ans. 4

5. Will the system run in an existing, heavily utilized operational environment?

Ans. 2

6. Does the system require online data entry?

Ans. 5

7. Does the online data entry require the input transaction to be built over multiple screens operations?

Ans. 3

8. Are the ILFs updated online?

Ans. 5

9. Are the inputs, outputs, files, or inquiries complex?

Ans. 1

10. Is the internal processing complex?

Ans. 4

11. Is the code designed to be reusable?

Ans. 1

12. Are conversion and installation included in the design?

Ans. 3

13. Is the system designed for multiple installations in different organizations?

Ans. 1

14. Is the application designed to facilitate change and ease of use by the user?

Ans. 4

TOTAL = 44

Once these data have been collected, a complexity value is associated with each count.

Organizations that use function point methods develop criteria for determining whether a particular entry is simple, average, or complex. To compute function points (FP), the following relationship is used:

$$FP = \text{count total} \times [0.65 + (0.01 \times \sum(F_i))]$$

where count total is the sum of all FP entries obtained from the table given below.

Information domain value	Count	Weighing factor			=	
		Simple	Average	complex		
External Inputs	15	X	3	4	6	= 45
External Outputs	7	X	4	5	7	= 28
External Inquiries	1	X	3	4	6	= 3
Internal Logical Files	4	X	7	10	15	= 28
External interface files	0	X	5	7	10	= 0
Count Total						= 104

In our project there are, we have the weighting factor as simple for all the five information domain values.

$$\begin{aligned}
 W(FP) &= \text{count total} \times [0.65 + (0.01 \times \sum(F_i))] \\
 &= 104 \times [0.65 + (0.01 \times 44)] = 104 \times 1.09 \\
 &= 113.36 \text{ FP}
 \end{aligned}$$

Let

Average Productivity, X = 4 FP/pm

Labour rate, Y = Rs 8000 person per month

So,

$$\text{Cost per FP} = Z = Y/X = \text{Rs } 8000/4 = \text{Rs } 2000(\text{approx.})$$

$$\text{Total estimated project cost} = W*Z$$

$$= 113.36 \times 2000$$

$$= \text{Rs } 2,26,720(\text{approx.})$$

$$\text{Estimated effort} = W/X$$

$$= 113.36/4$$

$$= 28 \text{ person-month (approx.)}$$

# EFFORT ESTIMATION USING COCOMO MODEL

Constructive Cost Model (COCOMO II) is a more comprehensive estimation model. COCOMO II is actually hierarchy of estimations models that address the following areas:

**Application composition model** – Used during the early stages of software engineering, when prototyping of user interfaces, consideration of software and system interaction, assessment of performance, and evaluation of technology maturity are paramount.

**Early design stage model** – Used once requirements have been stabilized and basic software architecture has been established.

**Post-architecture-stage model** – Used during the construction of the software. The COCOMO II models required sizing information.

Three different sizing options are available as part of the model hierarchy:

- Object points
- Function points
- Lines of source code

The object point is an indirect software measure that is computed using counts of the number of

- (1) screens (at the user interface),
- (2) reports, and
- (3) components likely to be required to build the application.

Each object instance is classified into one of the three complexity levels based on the following table –

**Complexity weighting for object types:**

OBJECT-TYPE	COMPLEXITY WEIGHT		
	Simple	Medium	Difficult
Screens	1	2	3
Reports	2	5	8

Reusable Components			10
---------------------	--	--	----

The object count is determined by multiplying the total number of object instances by weighting factor.

When component-based development or general software re-used is to be applied, the percent of reuse is estimated, and object count is adjusted:

$$\text{NOP} = (\text{object points}) \times [(100 - \% \text{ re-use})/100]$$

Where NOP is defined as new object points. To derive an estimate of effort based on the computed NOP value, a “productivity rate” must be derived.

$$\text{PROD} = \text{NOP} / \text{person-month}$$

Developer's experience or capability	Very Low	Low	Normal	High	Very High
Environment maturity or capability	Very Low	Low	Normal	High	Very High
Productivity	4	7	13	25	50

Once the productivity rate has been determined, an estimate of project effort is computed using –

$$\text{Estimated effort} = \text{NOP} / \text{PROD}$$

### COCOMO Estimation for our project –

Number of screens = 10

Number of reports = 2

Number of reusable components used = 0

In our project, there are simple screens and reports.

So,

$$\text{Object Point} = 10*1 + 2*2 = 14$$

Since we're not re-using any of the components in our project, the % re-use is zero here.

$$\text{NOP} = 14 * [(100-0)/100] = 14$$

$$\text{PROD} = 4$$

Now,

$$\text{Estimated Effort} = \text{NOP} / \text{PROD} = 14/4 = 3.5 \text{ person-month} = 4 \text{ person-month}$$

# RISK ANALYSIS

Risk analysis is a series of steps that help a software team to understand and manage uncertainty. A risk is a potential problem – it might happen, it might not.

## SOFTWARE RISKS

Risk always involves two characteristics:

- Uncertainty: the risk may or may not happen; that is, there are no 100% probable risks
- Loss: if the risk becomes a reality, unwanted consequences or losses will occur

## PHASES INVOLVED IN RISK ANALYSIS AND MANAGEMENT

- Risk Identification
- Risk Analysis
- Risk ranking and assessment
- Creating risk plan or RMMM plan

## TYPES OF RISKS

According to general categorization there are 3 types of risks:

- Known Risk
- Predictable Risk
- Unpredictable Risk

Another category of risk type:

- Project Risk
- Technical Risk
- Business Risk

### Risk identification

i.e., to recognize what can go wrong. Analysis of each risk to determine the likelihood that it will occur and the damage that it will do if it does occur.

**Product Size (PS):** Risk associated with overall size of the product to be built or modified.

**Business impact (BU):** Risk associated with constraints associated by the management or the marketplace.

**Customer characteristics (CU):** Risk associated with sophistication of the customer and developer's ability to communicate with the customer in a timely manner.

**Technology to be built (TE):** Risk associated with the complicity of the system. To be build and new risk of the technology that is packed by the system.

**Development Environment (DE):** Risk associated with the availability and quality of the tools to be used to build the product.

**Staff size and experiences (ST):** Risk associated with the overall technical and project experiences of the software engineers who will do the work.

**Process Definition (PD):** Risk associated with degree to which the software process has been defined and is followed by the developed organization

### **Assessing overall project risk**

The following questions have derived from risk data obtained by surveying experienced software project managers in different parts of world.

1. Have top software and customer managers formally committed to support the project? -**Yes**
2. Are end users enthusiastically committed to the project and the system/product to be built? –**Yes**
3. Are requirements fully understood by the software engineering team and its customers? -**Yes**
4. Have customers been involved fully in the definition of requirements? - **Yes**
5. Do end users have realistic expectations? -Yes Is the project scope stable? -**Yes**
6. Does the software engineering team have the right mix of skills? -**Yes**
7. Are project requirements stable? –**Yes**
8. Does the project team have experience with the technology to be implemented? -**Yes**
9. Is no. of people on the project team adequate to do the job? -**Yes**
10. Do all customers/user constituencies agree on the importance of the project and on the requirements for the system/product to be built? - **Yes**

## RISK TABLE FOR OUR PROJECT

RISKS	CATEGORY	PROBABILITY	IMPACTS
Size estimate may be significantly low	PS	20%	3
Large no. of user than planned	PS	10%	2
Less user than planned	PS	15%	2
End user may resist the system	BU	10%	2
Delivery deadline will be tightened	BU	10%	2
Funding will be lost	CU	15%	1
Customer will change requirements	CU	15%	3
Staff inexperienced	ST	20%	2
Technology will not meet expectation	TE	15%	4
Developing wrong user interface	DE	10%	2

**PS- Project size risk**

**Impact Values**

**BU- Business impact risk**

**1. Catastrophic**

**CU- Customer characteristics risk**

**2. Critical**

**DE- Development environment risk**

**3. Marginal**

**ST- Staff size and experience risk**

**4. Negligible**

**TE-Technology to be built**

**PD-Process Definition**

# TESTING

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest.

The purpose of software testing is to identify bugs or errors, gaps or missing requirements in contrast to actual requirements. It also involves evaluating the features of the software product for requirements in terms of security, reliability and performance.

Software testing is one element of a broader topic that is often referred to as verification and validation.

**WHITE-BOX TESTING:** White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing or Glass box testing.

**BASIS-PATH TESTING:** Basis Path Testing is a White Box Testing method in which test cases are defined based on flows or logical paths that can be taken through the program. The objective of basis path testing is to define the number of independent paths, so the number of test cases needed can be defined explicitly to maximize test coverage. It involves execution of all possible blocks in a program and achieves maximum path coverage with the least number of test cases.

We are performing white-box testing for Login module.

PSEUDOCODE FOR LOGIN:

(1)Begin:

(2)LoginRequest()

(3) Username=GetUsername()

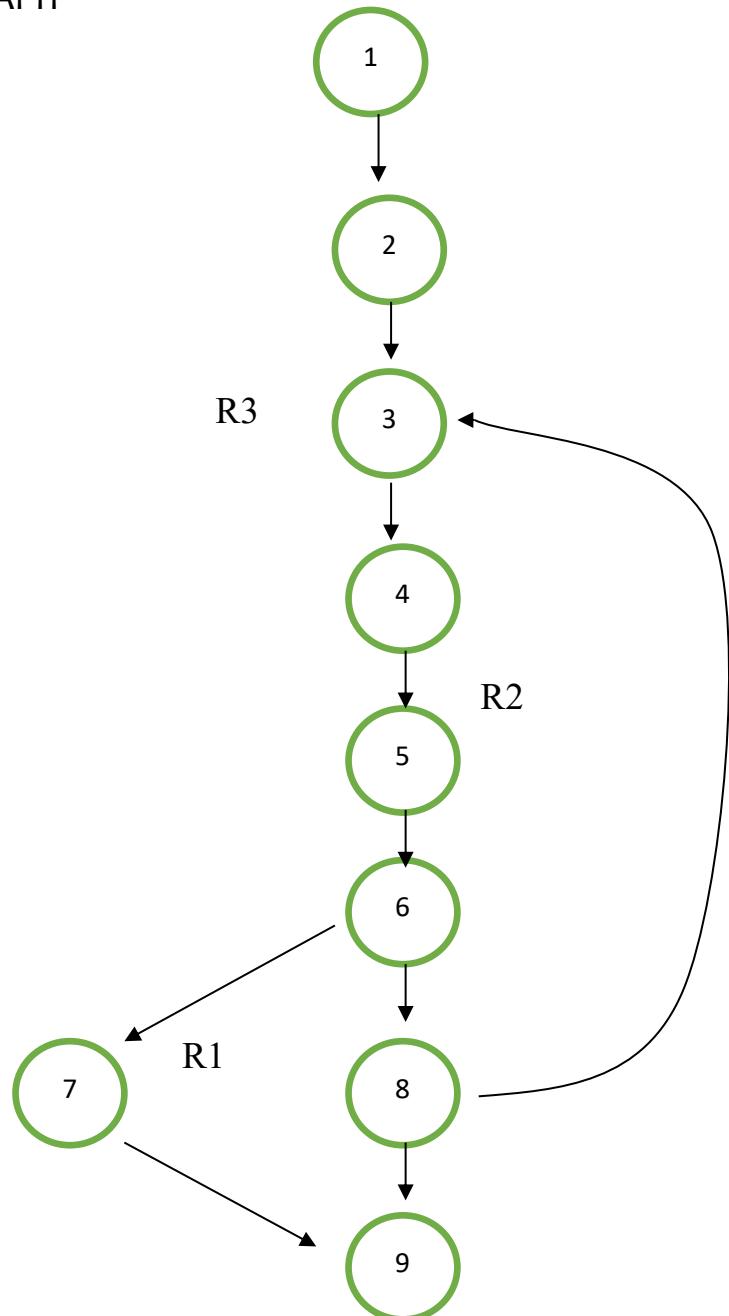
(4) Password=GetPassword()

(5) for (i = 0; i < UsersInDB; i++)

(6) if (Entered\_Username == Username[i] &&  
Entered\_Password == Password[i])

```
{(7) print("welcome 'username' ")  
    return TRUE }  
  
(8) print ("Incorrect username or password!!")  
    return FALSE  
  
(9) End
```

FLOW GRAPH



### CYCLOMATIC COMPLEXITY OF RESULTING GRAPH:

1.  $V(G) = \text{Number of regions of the flow graph} = 3$

2.  $V(G) = E - V + 2$

$$= 10 - 9 + 2$$

$$= 3$$

Where  $E = \text{Number of edges in the flow graph}$  and

$N = \text{Number of nodes in the flow graph}$

3.  $V(G) = P + 1$

$$= 2 + 1 (\text{6}^{\text{TH}} \text{ AND } \text{8}^{\text{TH}} \text{ NODE})$$

$$= 3$$

Where  $P = \text{Predicate nodes}$

Hence by all means cyclomatic complexity is 3.

### BASIS SET OF LINEARLY INDEPENDENT PATHS:

PATH 1: 1-6, 7, 9

PATH 2: 1-6, 8, 9

PATH 3: 1-6, 8, 3-6, 8, 9

### TEST CASES:

TEST CASE ID	INPUT	ACTUAL OUTPUT	EXPECTED OUTPUT
1	Incorrect username and password	To be observed after execution	Incorrect username or password!!
2	Correct username and password	To be observed after execution	Welcome "username"

3	Incorrect username	To be observed after execution	Incorrect username or password!!
4	Incorrect password	To be observed after execution	Incorrect username or password!!

## BIBLIOGRAPHY

1. Software Engineering – A Practitioner’s Approach – Roger S. Pressman.
2. Aggarwal , K.K., & Singh,Y.(2007) . Software Engineering 3<sup>rd</sup> edition. New Age International Publishers.
3. IEEE Software Engineering Standards Committee, “IEEE Std. 29148:2011, IEEE Recommended Practice for Software Requirements Specification.
4. Software Engineering by Rajib Mall.