

Mercy Application



Analyzed and built by

- Manar Altairy
- Jeelan Alotaibi
- Elaf Aloufi
- Shatha Alfaifi
- Reema Alghamdi

Supervised by

- Dr.Taghreed Bagies

Topics

01

Problem
Definition

02

Solution

03

Approach

04

Tools

05

Process

06

Functional
Requirements

07

Non-functional
Requirements

08

Use case
Diagram

09

Class
Diagram

10

Sequence
Diagram

11

Interfaces
Design

12

Running
Code

13

Testing

14

Challenges


15

Conclusion &
Future Work



01

Problem definition

- Donate
 - Ask for a donation
 - Adoption
 - Offer Pets for adoption
 - Help stray animals
- 

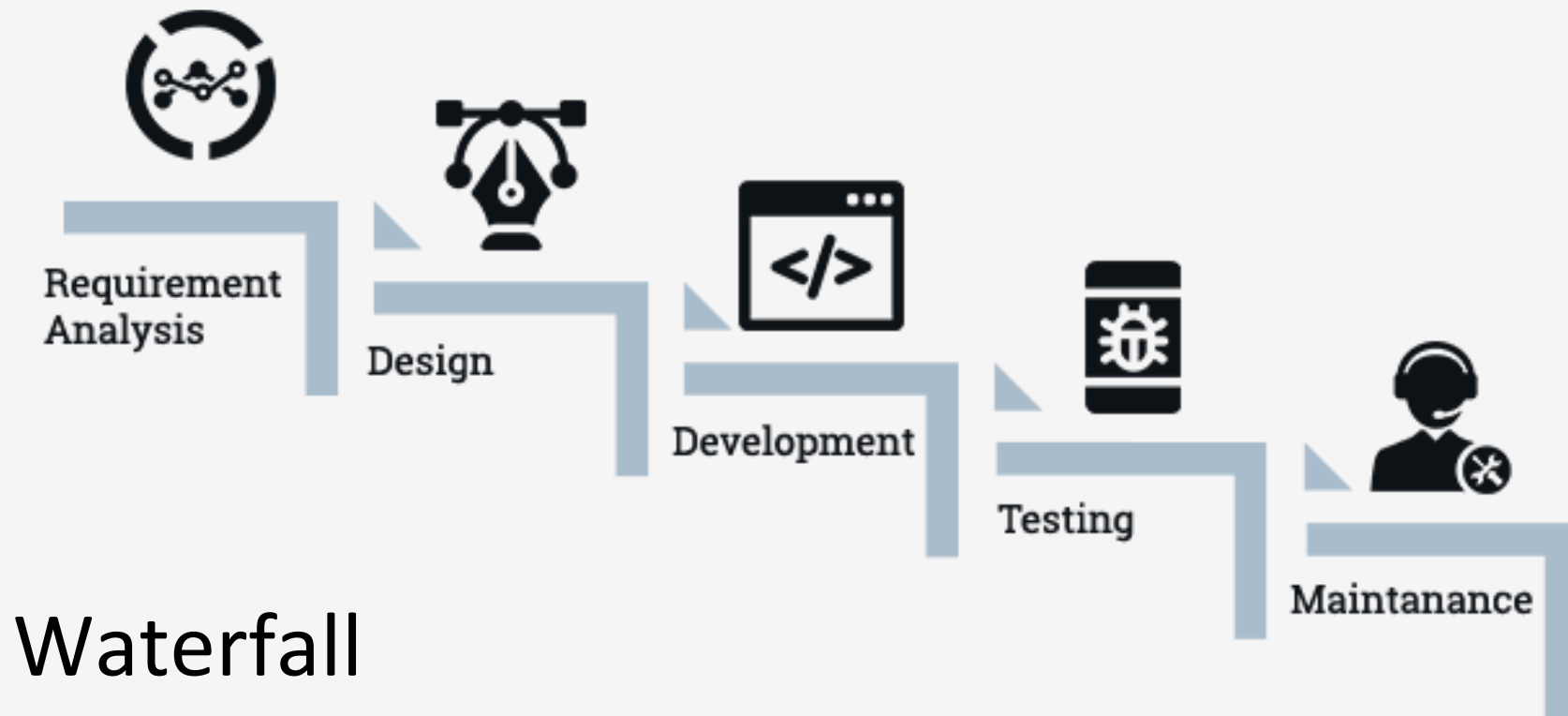
02

Solution



03

Our Approach





Project



Drive



Lucidchart



Google Docs

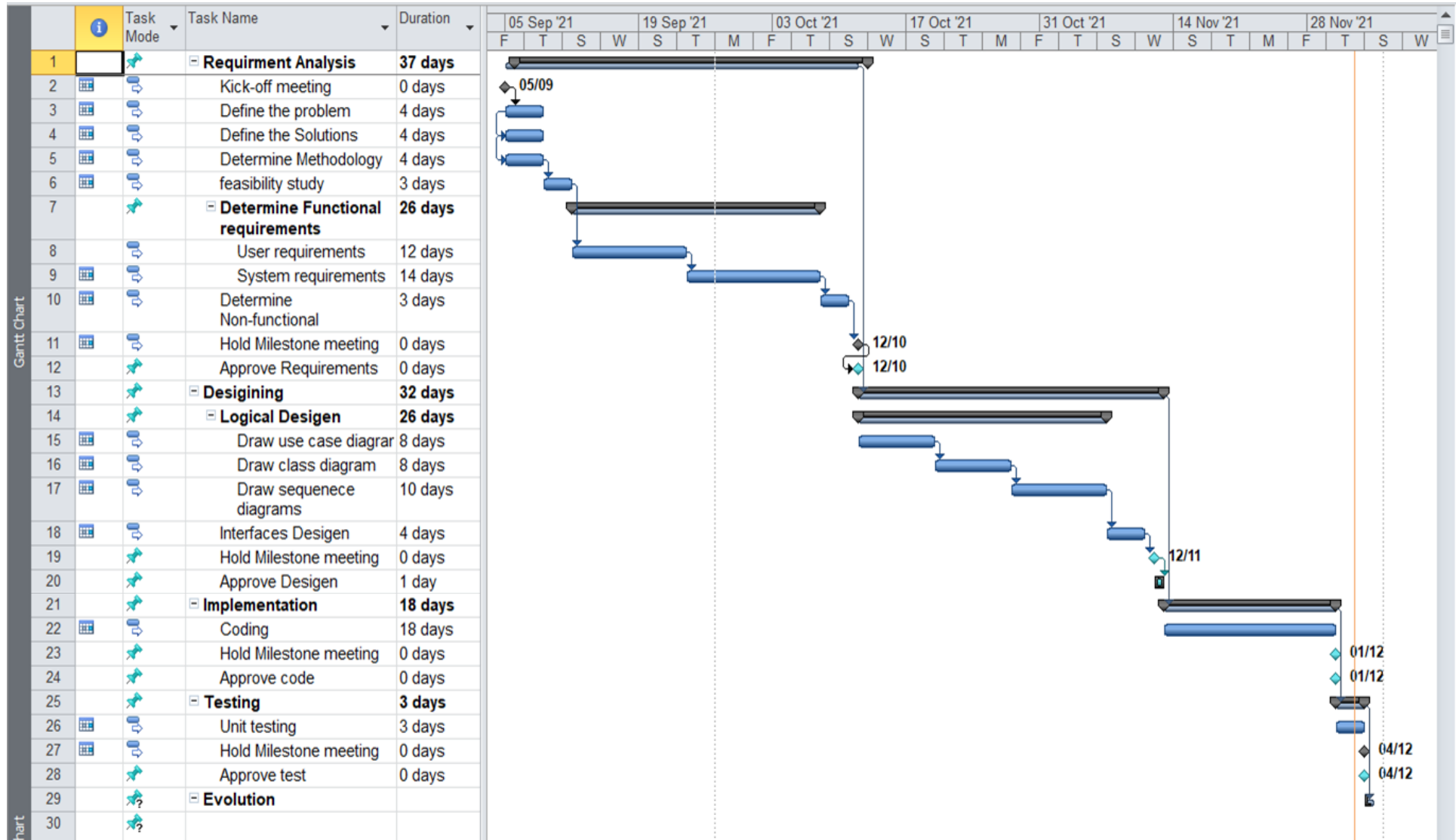


Google Slides



04
Tools

05 Process



06

Functional Requirements

User Requirements

- The user shall be able to register as pet owner or a customer
- The user shall be able to login by username and password
- The user shall be able to chat with other users
- The user shall be able to report stray pets
- The customer shall be able to donate money to pet owners from donation request
- The customer shall be able to donate supplies to pet owners from donation request
- The pet owners shall be able to acquire donated supplies from the customers
- The pet owner shall be able to request supply donation by filling a form
- The pet owners shall be able to confirm donated supplies from customers

- The system shall allow users to register only as pet owner user or customer user
- The system shall only accept unique username from user
- The system shall verify user password before login
- The system shall display features once the user login by username and password
- The system shall alert the pet owner by notification once a customer donates a supply
- The system shall update available offered pet for adoption page, once a pet owner confirms the adoption of the pet
- The system shall send the adoption form to the pet owner once the customer submits the form
- The system shall open a chat between the pet owner and the customer once the pet owner accepted the pet adoption request

06

Functional Requirements

System Requirements



Portability



Usability



Security



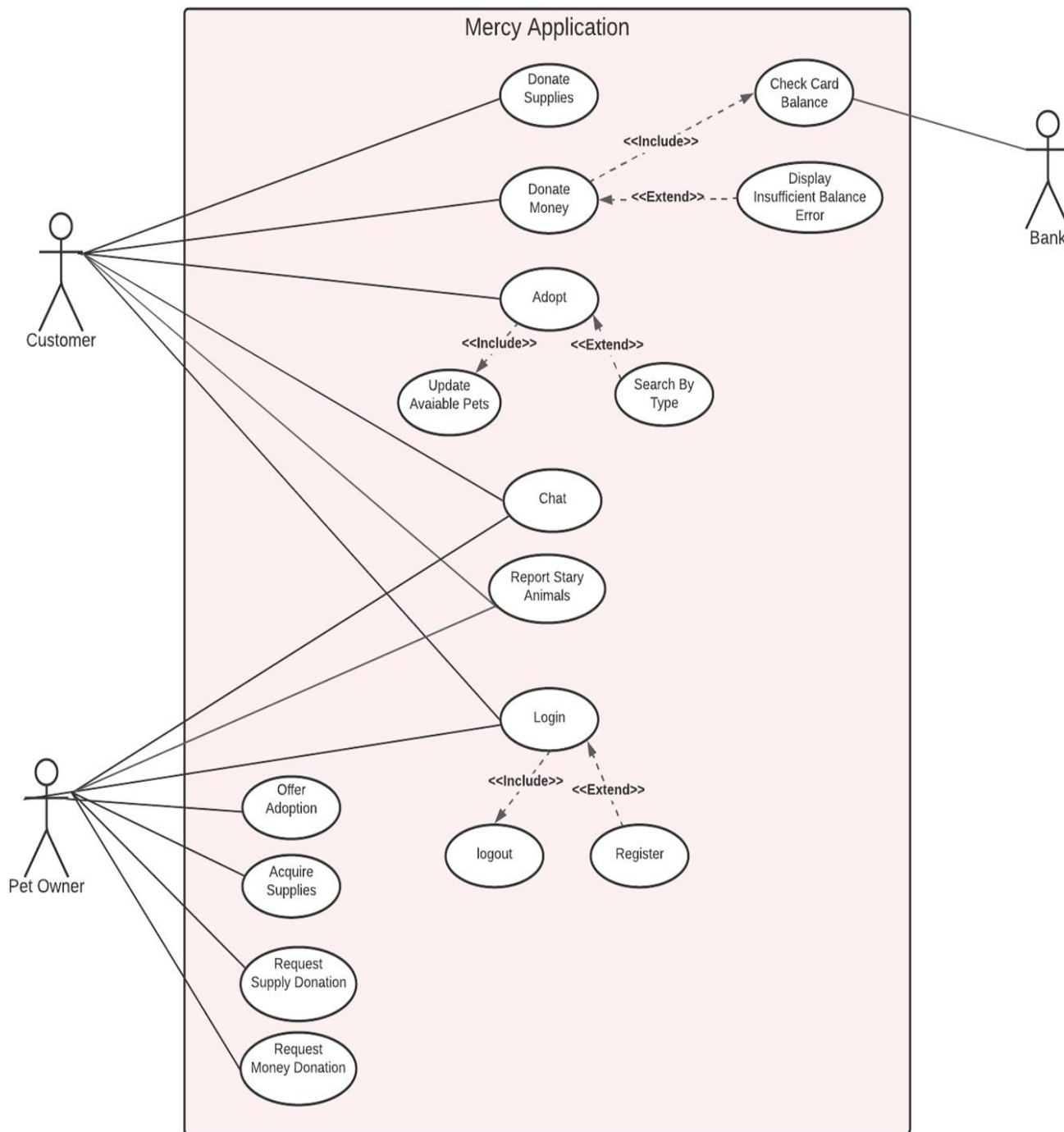
Performance



Maintainability

07

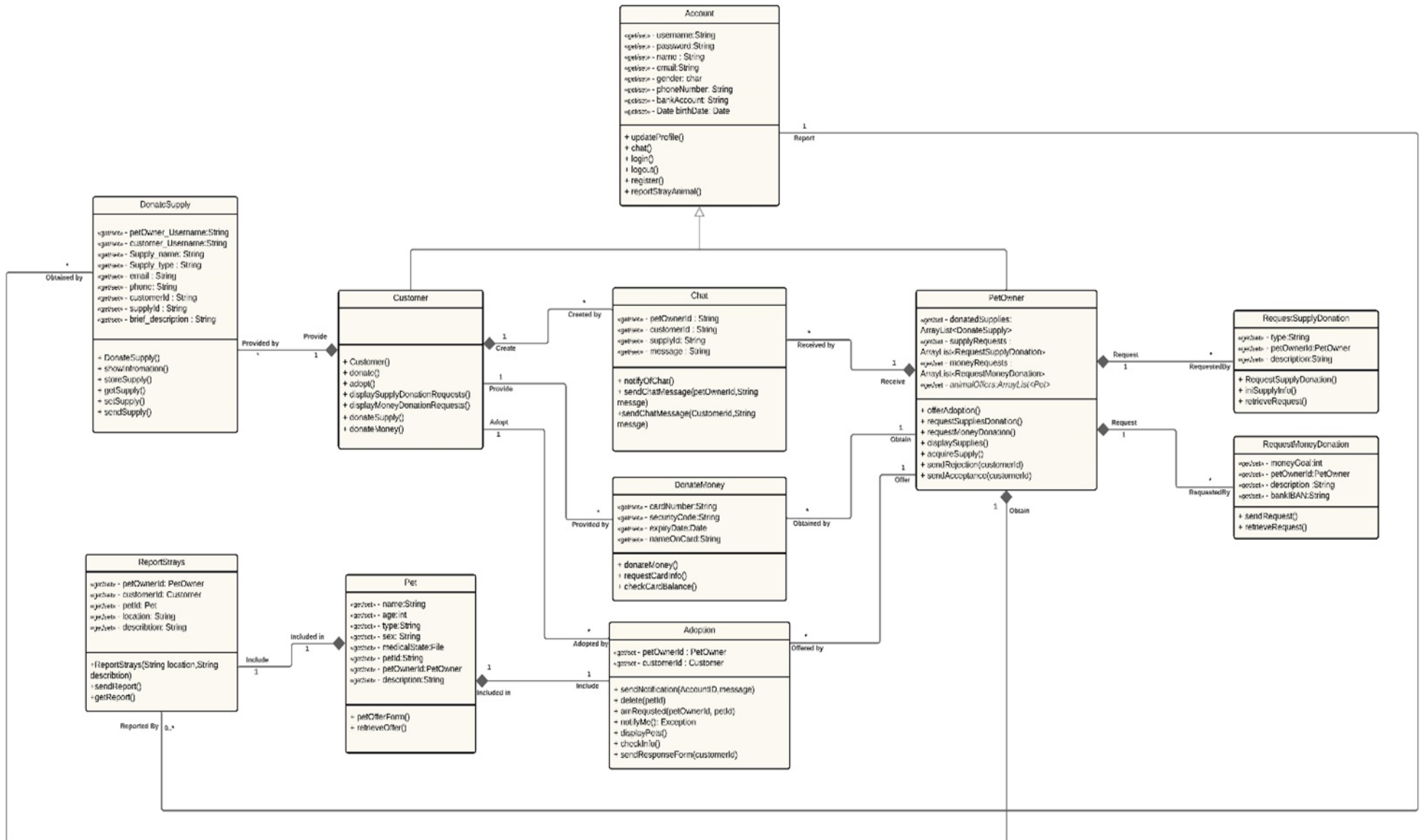
Non Functional Requirements



08

Use case Diagram

09 Class Diagram

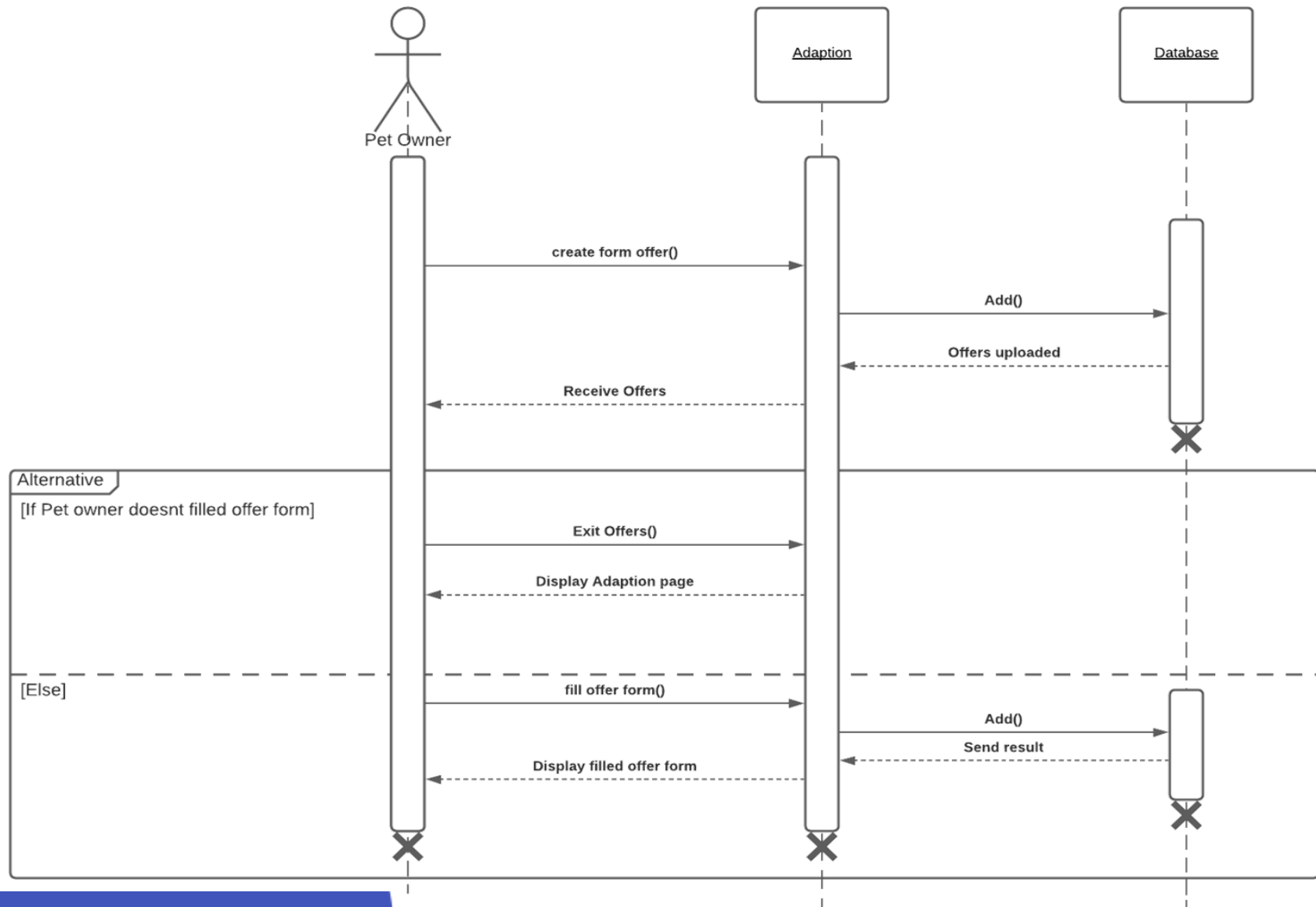




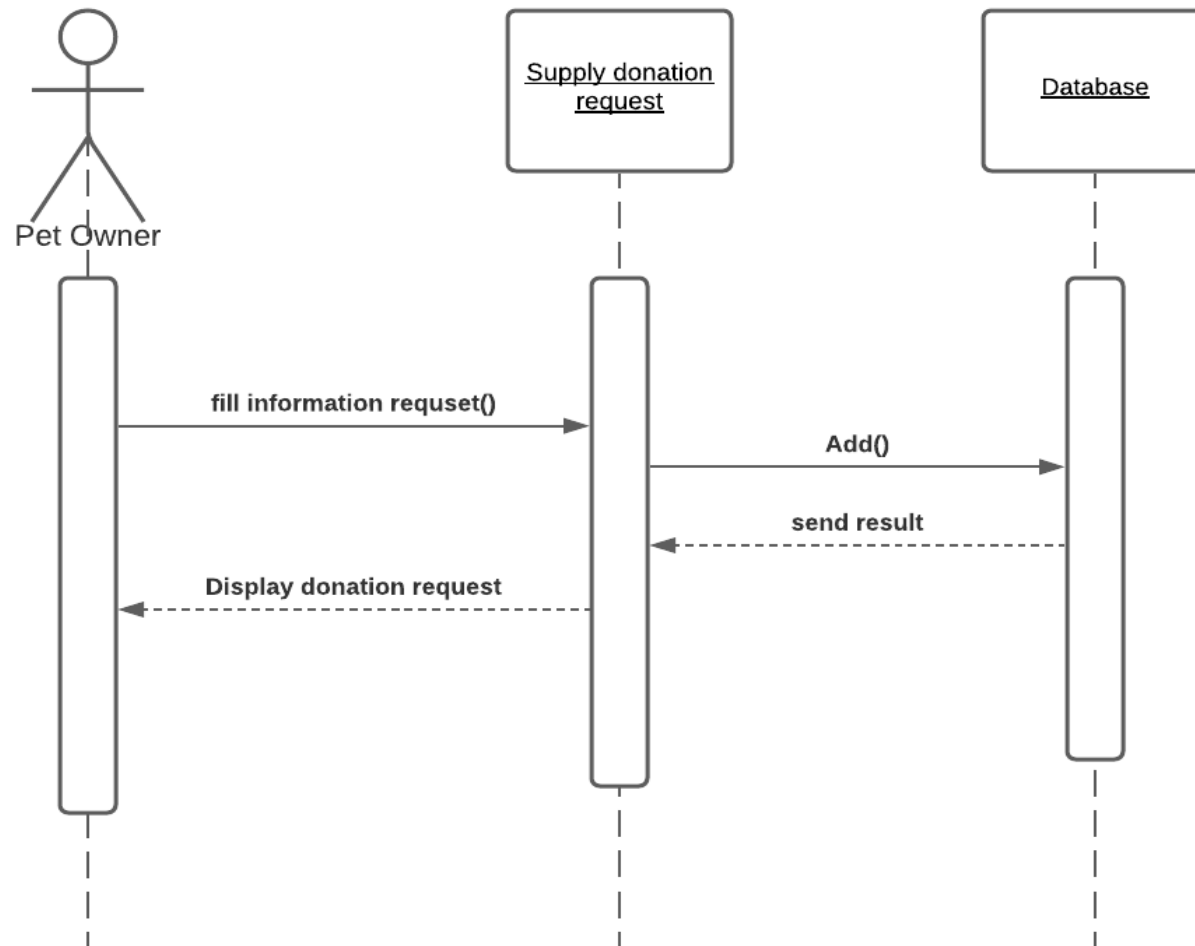
10

Sequence Diagrams

Offer Adoption



Request Supply Donation

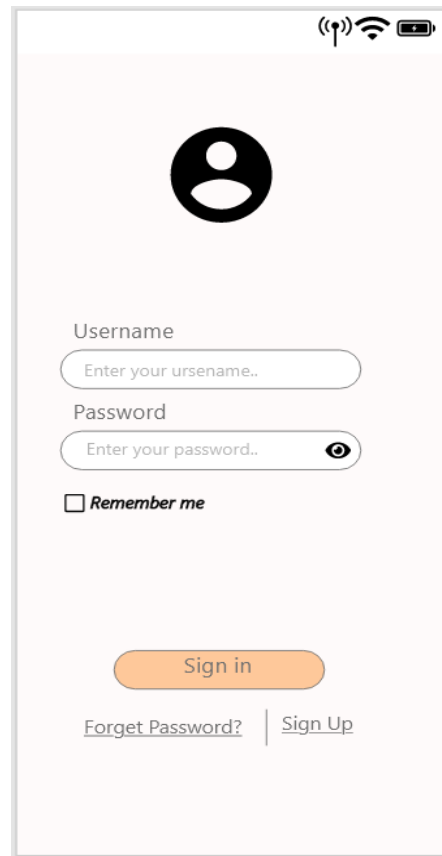




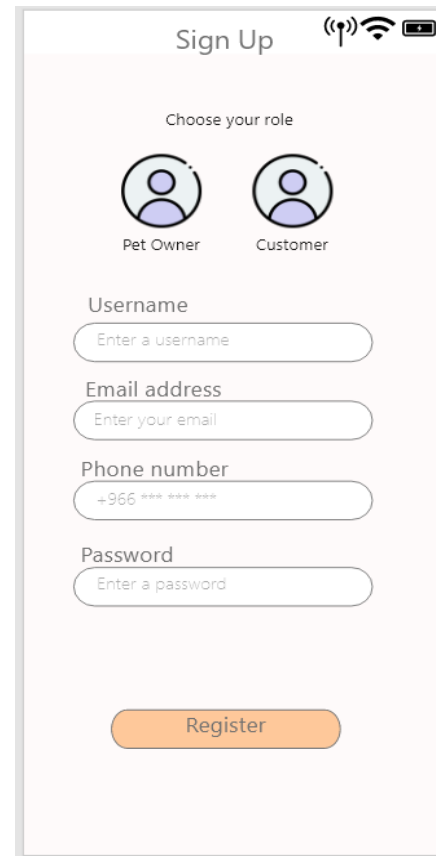
11

Interfaces Design

- Login/Register Interfaces:

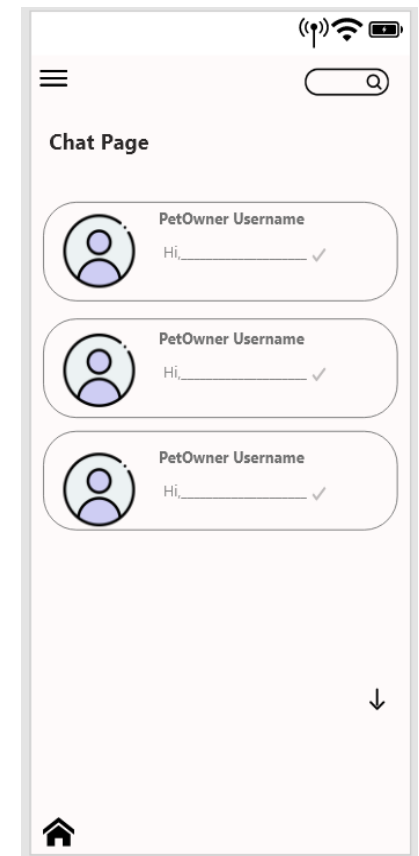
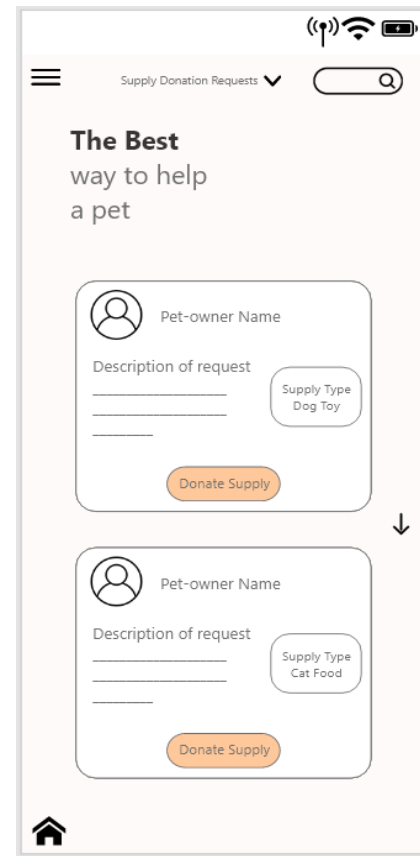
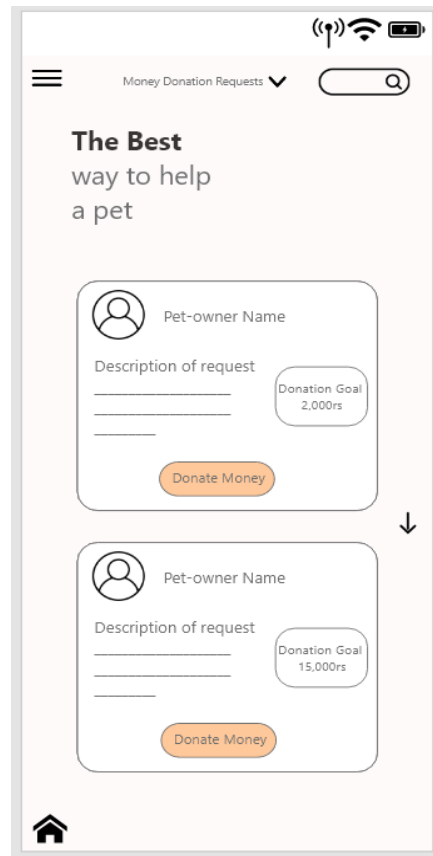
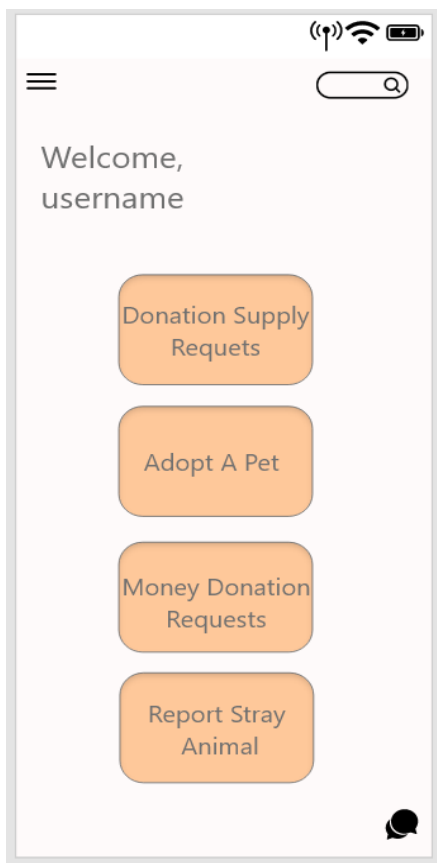


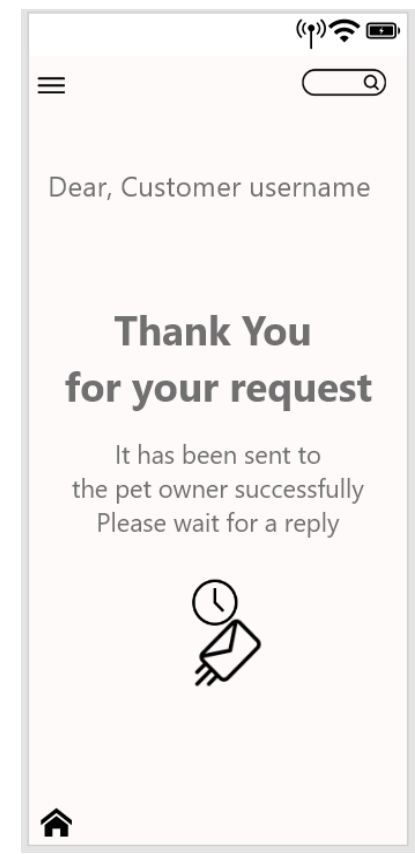
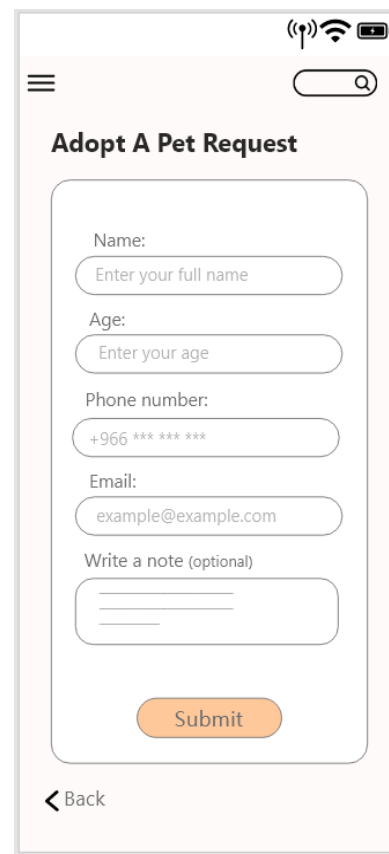
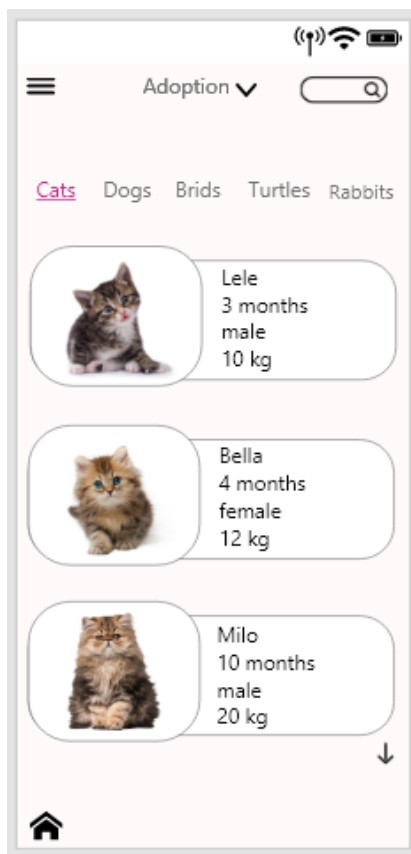
Mobile app login interface. At the top, there is a status bar with signal, Wi-Fi, and battery icons. Below it is a large black circular icon representing a user profile. The form contains two input fields: 'Username' with placeholder text 'Enter your username..' and 'Password' with placeholder text 'Enter your password..' and an eye icon for toggling visibility. Below the password field is a checkbox labeled 'Remember me'. At the bottom, there is an orange 'Sign in' button and two links: 'Forget Password?' and 'Sign Up'.



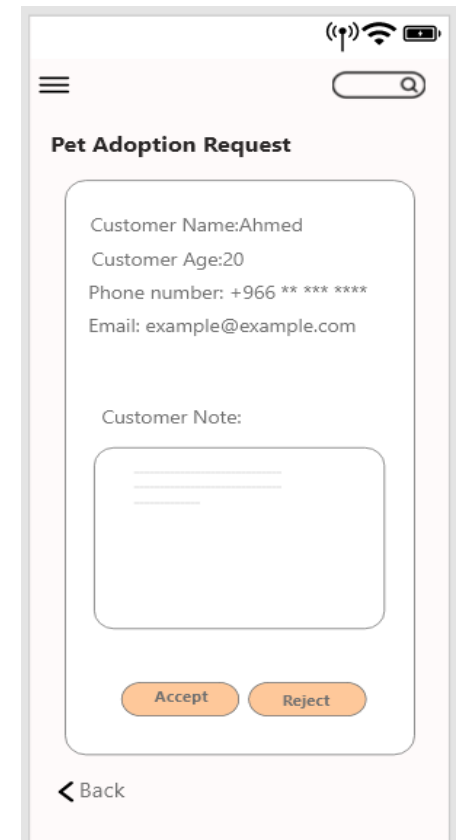
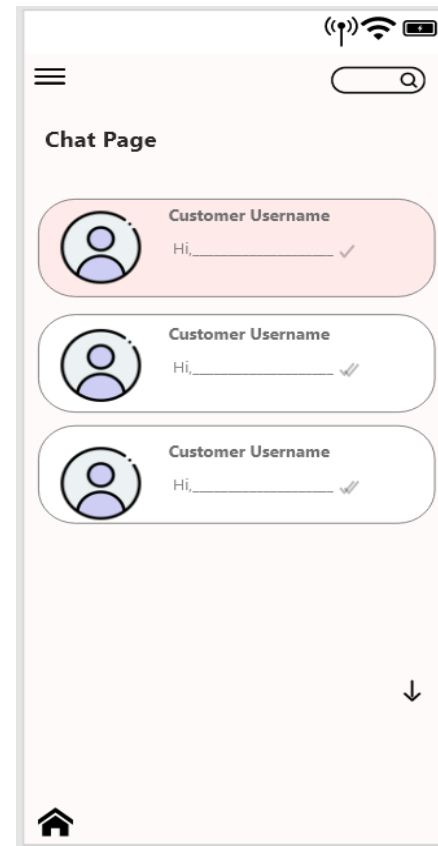
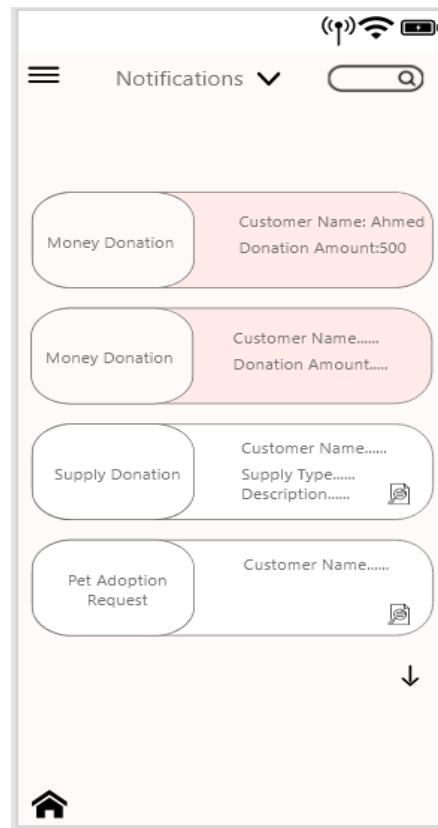
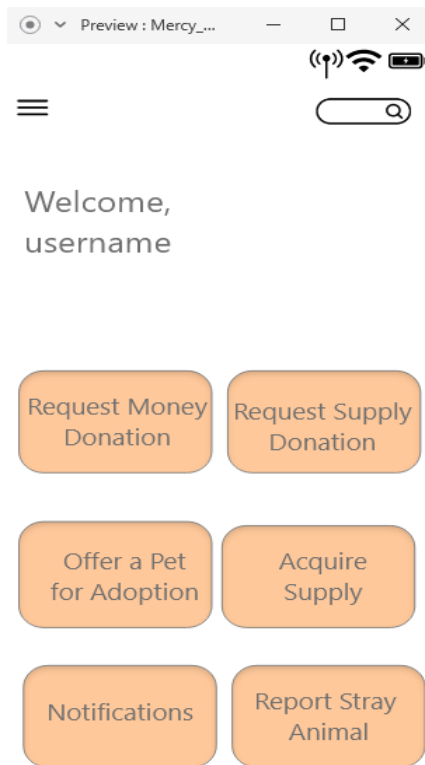
Mobile app sign up interface. At the top, there is a status bar with signal, Wi-Fi, and battery icons. Below it is the title 'Sign Up'. The form starts with a 'Choose your role' section featuring two circular icons: 'Pet Owner' and 'Customer'. Below this are four input fields: 'Username' (placeholder: 'Enter a username'), 'Email address' (placeholder: 'Enter your email'), 'Phone number' (placeholder: '+966 *** **'), and 'Password' (placeholder: 'Enter a password'). At the bottom, there is an orange 'Register' button.

• Customer Interfaces:





• Pet Owner Interfaces :



Acquire Supply

Customer Name:Nawaf
Supply Type:food.
Phone number: +966 ** *
Email: example@example.com
Supply Picture:

Customer Note:

Accept

Reject

Offer adoption

Upload
a photo

Pet name:

enter your pet name

Age:

eg. 2

Type:

eg. dog

Upload medical passport *

Contact information

Phone number

+966 ** *
Email

example@example.com

Submit

Offer adoption

Dear, pet Owner Name

Your adoption offer
has been submitted
successfully

Preview : Mercy_...

Request Money Donation

Name:
Enter your name

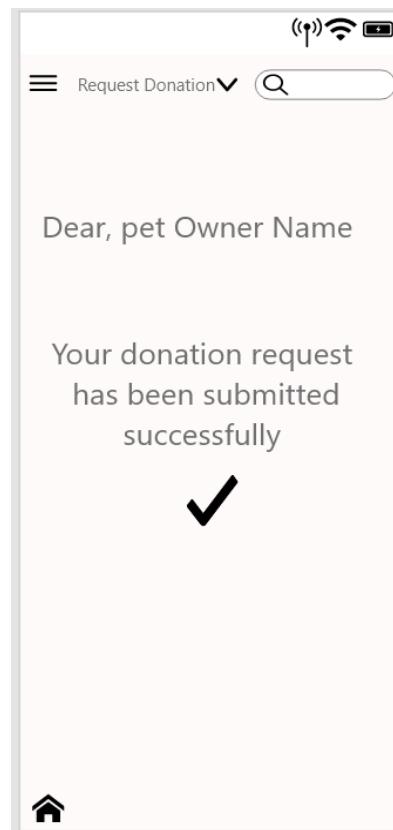
IBN
SA-

Enter Donation Goal:
0rs

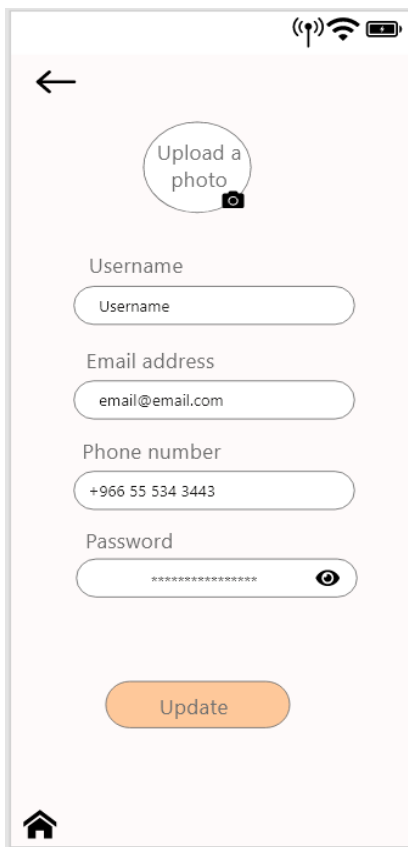
Write a Description

Submit

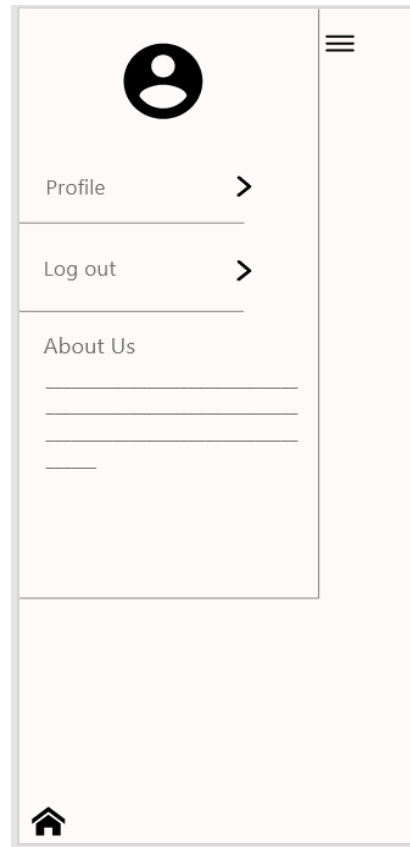
< Back



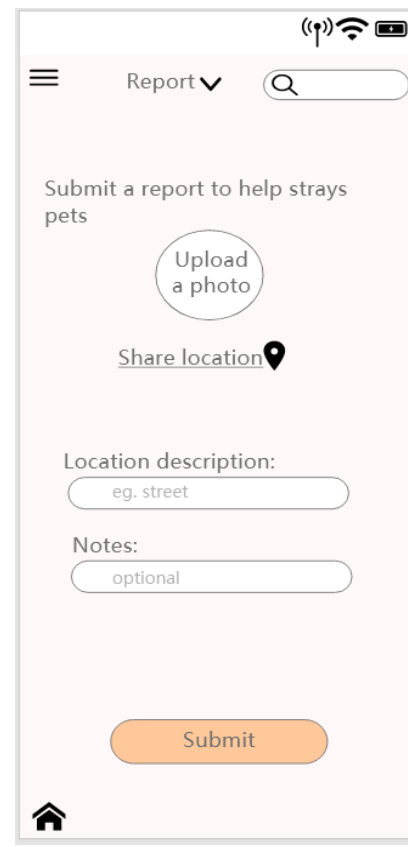
• Shared Interfaces :



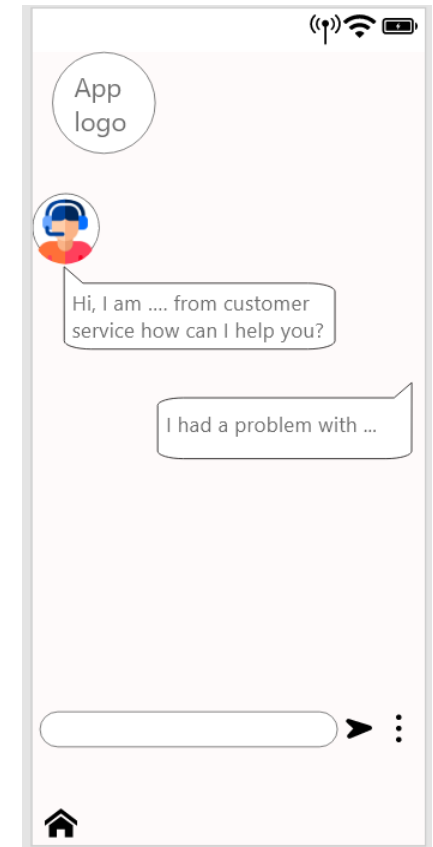
Mobile app registration screen. At the top, a back arrow and status icons (signal, Wi-Fi, battery) are visible. Below the back arrow is a circular button labeled "Upload a photo" with a camera icon. The form contains four input fields: "Username" (placeholder: Username), "Email address" (placeholder: email@email.com), "Phone number" (placeholder: +966 55 534 3443), and "Password" (placeholder: ***** with an eye icon for toggling visibility). An orange "Update" button is at the bottom. A home icon is in the bottom left corner.



Mobile app profile screen. At the top, a user profile icon and a hamburger menu icon are visible. Below the profile icon is a "Profile" section with a right arrow. Below that is a "Log out" section with a right arrow. The "About Us" section contains three horizontal lines for text input. A home icon is in the bottom left corner.



Mobile app report screen. At the top, a hamburger menu icon, the text "Report" with a dropdown arrow, and a search bar are visible. Below the search bar is the text "Submit a report to help strays pets". This is followed by a circular "Upload a photo" button and a "Share location" button with a location pin icon. Below these are two input fields: "Location description:" (placeholder: eg. street) and "Notes:" (placeholder: optional). An orange "Submit" button is at the bottom. A home icon is in the bottom left corner.



Mobile app chat screen. At the top, status icons (signal, Wi-Fi, battery) are visible. Below them is a circular "App logo" button. A chat bubble from a user (represented by a profile icon) says "Hi, I am from customer service how can I help you?". Below this is a response bubble that says "I had a problem with ...". At the bottom, there is a text input field, a right arrow, and a vertical ellipsis menu icon. A home icon is in the bottom left corner.



12

Running Code

Running Code

```
-- you have to log in to use any feature in the system --  
  
----- Log in -----  
  
Your Username : Reem10  
Your password : 1235  
  
.... Wrong password or username!!! try again ....  
  
-- you have to log in to use any feature in the system --  
  
----- Log in -----  
  
Your Username : Reemal0  
Your password : 1234  
  
----- Successfully log in -----  
  
.....Welcome Pet owner.....  
  
1. Request supply donation  
2. Offer adoption  
3. Acquire Supply  
4. log out  
  
Choose a function : |
```

-- you have to log in to use any feature in the system --

----- Log in -----

Your Username : Reema10

Your password : 1234

----- Successfully log in -----

.....Welcome Pet owner.....

1. Request supply donation
2. Offer adoption
3. Acquire Supply
4. log out

Choose a function : 1

-----Request Donation Form-----

Enter Your Supply Type: food

Write a Description: I can't effort food for my cat, cat food is n't available in my city these days I live in ...

Dear, Reema

Your donation request has been submitted successfully

.....Welcome Pet owner.....

1. Request supply donation
2. Offer adoption
3. Acquire Supply
4. log out

Choose a function : 4

... Thank you for using our app ...

-- you have to log in to use any feature in the system --

----- Log in -----

Your Username : |

Your Username : JeeOt
Your password : 1234

..... Successfully log in

.....Welcome customer.....

1. donate
2. adopt
3. log out

Choose a function : 1

-----Donation Requests-----
....The Best way to help a pet....

Choose Donation Type:

1. Display Money Donation Requests(not Available)
2. Display Donate Supply Donation Requests

Enter your choice: 2

=====

1.
Pet Owner Name: Shatha
Description of request:
I need cat toys, any toy would be helpful
Supply Type: Toy

=====

2.
Pet Owner Name: Manar
Description of request:
I need dog for my 5 month old dog
Supply Type: Food

=====

3.
Pet Owner Name: Reema
Description of request:
I can't effort food for my cat, cat food is n't available in my city these days I live in ...
Supply Type: food

Type (quit) to exit or choose a number of the pet owner to donate: 3

-----Donate Supply Form-----

Please Fill supply information

Enter supply type: food
Enter supply name: Meow Mix
Enter supply description: mix of chicken , turky and ocean fish
Enter your Email: JeeO@gmail.com
Enter your Phone Number: 0599866666

Dear, Customer Jeelan
Thank You for your donation
Your donation submitted successfully

Code

```
public class Home {

    public static Map<String, PetOwner> donationSupplyRequets = new HashMap<String, PetOwner>();
    public static Map<String, PetOwner> petOffers = new HashMap<String, PetOwner>();
    public static HashMap<String, PetOwner> PetOwners = new HashMap<String, PetOwner>(); //contains all petowner accoounts
    public static Map<String, Customer> Customers = new HashMap<String, Customer>(); //contains all customer accoounts

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        //Initialize system data + initialize users
        createPetOwnerUsers();
        createCustomerUsers();
        initializeDonationRequestInfo();
        initializeanimalOffersInfo();
        //////////////////////////////////////
```

Code

```
do {
    System.out.println("\n.....Welcome Pet owner.....\n");
    System.out.println("1. Request supply donation\n2. Offer adoption\n3. Acquire Supply\n4. log out\n");
    System.out.print("Choose a function : ");
    command = s.nextLine().trim();
    switch (command) {
        case "1":
            petowner.requestSuppliesDonation();
            break;
        case "2":
            petowner.offerAdoption(s);
            break;
        case "3":
            petowner.acquireSupply();
            break;
    }
} while (!command.equalsIgnoreCase("4"));
System.out.println("\n\n ... Thank you for using our app ... \n ");
```

Code

```
public class RequestSupplyDonation {  
  
    String type;  
    PetOwner petOwnerId;  
    String description;  
  
    public RequestSupplyDonation() {  
  
    }  
  
    public RequestSupplyDonation(PetOwner username) {  
        this.petOwnerId = username;  
        Scanner s = new Scanner(System.in);  
        System.out.println("\n-----Request Donation Form-----\n");  
        System.out.print("Enter Your Supply Type: ");  
        this.type = s.nextLine().trim();  
        System.out.print("Write a Description: ");  
        this.description = s.nextLine().trim();  
    }  
}
```

Code

```
////////// methods //////////  
public void requestSuppliesDonation() {  
    RequestSupplyDonation a = new RequestSupplyDonation(this);  
    supplyRequests.add(a);  
    Home.donationSupplyRequets.put(this.getUsername(), this);  
    System.out.print(" \nDear, " + this.getName()  
        + "\n Your donation request has been submitted successfully\n"  
        + "\n-----");  
}
```

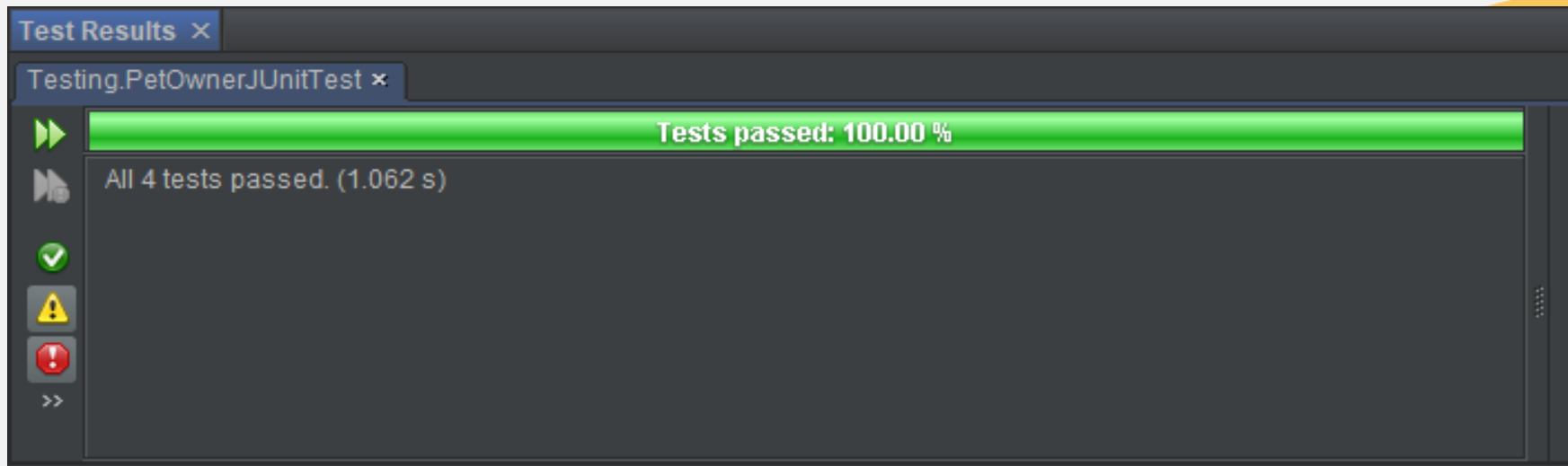

The background features abstract, organic shapes in blue and yellow. A large blue shape is in the bottom-left corner, and a yellow shape is in the top-right corner.

13

Testing

Testing Code

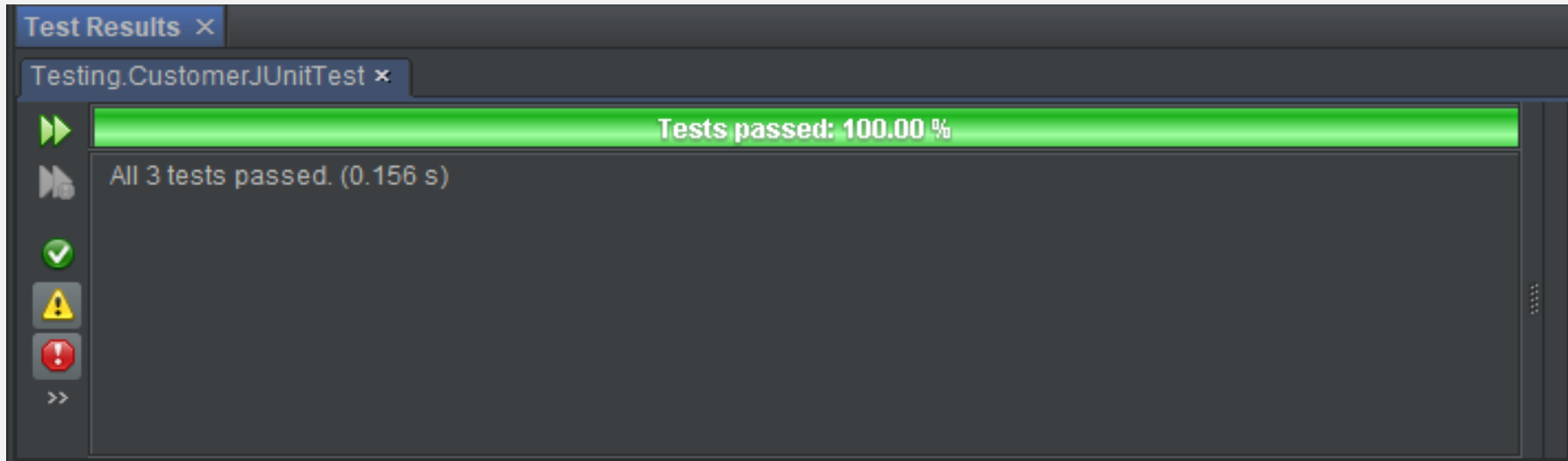
Pet owner class:



- reqSupplyDonationTest():To check if the method add the donation supply request in the donationSupplyRequets arrayList
- offerAdoptionTest(): Check if method successfully add the pet in the petOffers arraylist
- addDonatedSuppliesTest():To ensure that the method add the supply that donate by customer to donatedSupplies arrayList
- acquireSupplyTest():check if method display all supplies that donated by customer to pet owner

Testing Code

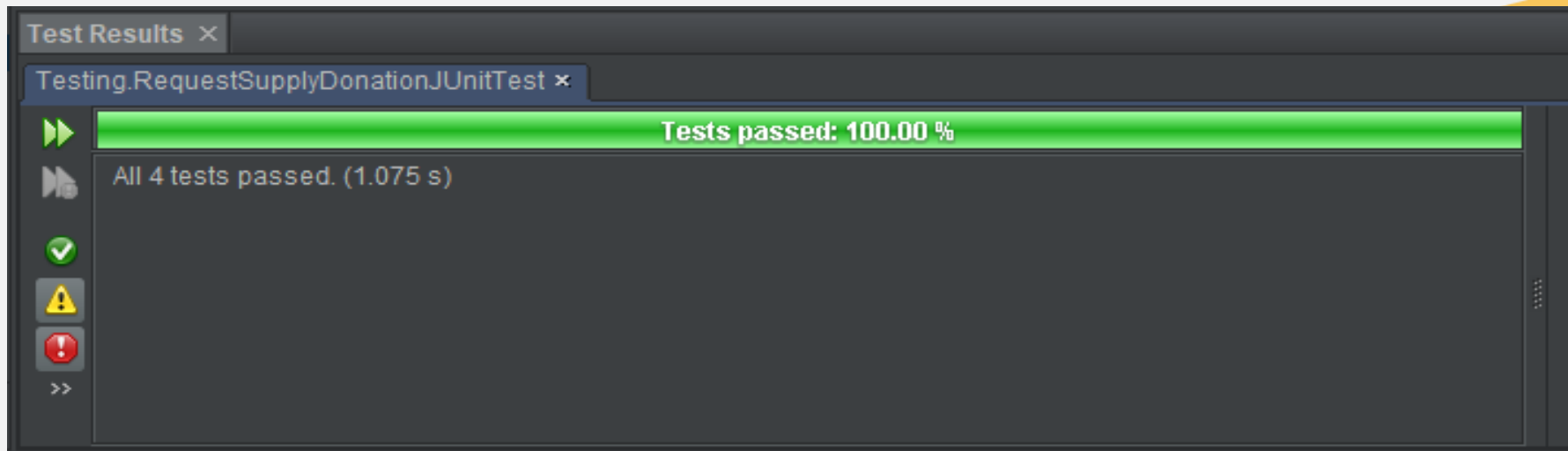
Customer class:



- Adopttest(): to ensure that all offers of pet adoption are displayed to the customer
- displaySupplyDonationRequeststest():to ensure that all donation request are displayed to the customer
- DonateSupplyTest(): To check if the donated supply is added into the pet owner request successfully or not

Testing Code

Request Supply Donation class:



- retrieveRequestTest():ensure that the method return the right information of supply request
- getTypeTest():ensure that the method return the type of the supply
- getDescriptionTest():ensure that the method return the supple description of the supply

14 Challenges

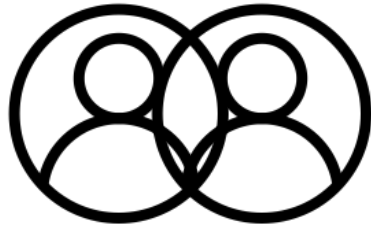




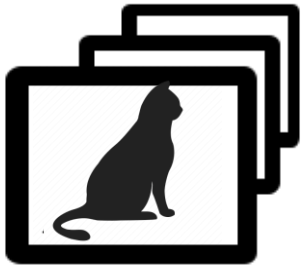
15

Conclusion

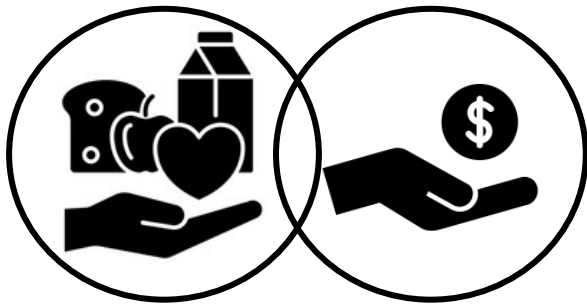
Every animal deserves to be happy, to have good owners and to live in a suitable environment.



Merging pet owner and
customer account



Add gallery of animals in
the user account



Merging the two types of
requests into one feature



open a shop within the
application featuring
offered pets and supplies
to buy

15

Future works



Thank You
For listening