Software Requirement Specification

AdoptMe

Francesca Pavone Salafia, Federica Cantelmi

Progetto Ingegneria del Software e Progettazione Web, A.A. 2022-23

Contents

1 INTRODUCTION 3

1.1 AIM OF THE DOCUMENT 3
1.2 OVERVIEW OF THE DEFINED SYSTEM 3
1.3 OPERATIONAL SETTINGS 3
1.4 RELATED SYSTEM 4
1.5 VOCABULARY 5

2 USER STORIES 7

3 FUNCTIONAL REQUIREMENTS 8

4 USE CASES 9

4.1 DIAGRAM 9
4.2 INTERNAL STEPS 10

5 STORYBOARDS 11

6 CLASS DIAGRAM 18

6.1 VOPC 18
6.2 MVC 20
6.3 DESIGN PATTERN 21

7 ACTIVITY DIAGRAM 22

8 SEQUENCE DIAGRAM 24

9 STATE DIAGRAM 25

10 TESTING 26

1 Introduction

1.1 Aim of the document

The aim of the document is to provide the functional specifications and product design of the system developed for the software engineering and web design course.

1.2 Overview of the defined system

AdoptMe's purpose is to ensure more visibility to all the pets staying in shelters and to help common people find pets that are more compatible with them, for the purpose of avoiding animal abandonment and ensuring a good quality of life to our four-legged friends.

AdoptMe can be used by two types of users.

As a shelter manager, you can sign up as a shelter in order to load all the information required to reach your facility and a list of all your pets that are currently adoptable, by adding all of their information.

As a person interested in adopting a pet, you can sign up as a user and load all your information or you can proceed without signing in or logging in.

A generic user (logged or not logged) can search for pets by entering a city or a specific shelter's name or by answering a questionnaire that will return a list of all the pets that are more compatible with the answers given.

If you decide to login, you will also be able to:

- save all the pets returned by the research, that will be kept in a specific area of the profile;
- make a request to meet a specific pet. After you've sent the request, you will arrange the ultimate date and time of the appointment with the shelter manager by accepting or modifying the request. If you decide to not meet the pet, you can also annul the request anytime you want.

In this system, you can interact with the software through two interfaces: one graphic and the other one by command line.

1.3 Operational settings

Required:

- Internet connection.
- MySQL connection.
- Maven installed and set.

1.4 Related system

Empethy

PROS:

- In order to send a request, it provides the pre-adoption form, which is mandatory for the adoption, and proceeds with the immediate submission to the shelter.
- It maintains a blog where you can find suggestions to take care of your pet.

CONS:

- You can’t login with Google.
- It doesn’t maintain information about the shelter.
- It provides few filters for the research of a pet.

AmazDog

PROS :

- The platform offers, in addition to search for adoptable pets, the possibility to look for beaches and hotels where dogs are accepted.
- It maintains an area dedicated to lost pets, where you can post your advert.

CONS:

- You can’t do anything without logging in.

1.5 Vocabulary

- Information:
    - Type: dog, cat
    - Photo
    - Name
    - Date of birth (if not known: indicative month and/or year of birth)
    - Gender: male, female
    - Size (for dogs): small, medium, large, extra large
    - Coat length: short, medium, long
    - Disability: yes, no (if yes: type of disability)
    - Test fiv (for cats): positive, negative
    - Test felv (for cats): positive, negative
    - Vaccinated: yes, no
    - Microchipped: yes, no
    - Dewormed: yes, no
    - Sterilized: yes, no
    - Need of a program of dog education (for dogs): yes, no
    - Compatibility: male dogs, female dogs, male cats, female cats, children, elders, first experience, sleeping outside, 1-3 / 4-6 / more than 6 hours a day alone

- Main information:
    - Photo
    - Name
    - Gender: male, female
    - Age: puppy (0-12 months), young (13 months-3 years), adult (4-10 years), senior (more than 11 years)

- Questionnaire:
    - “Which type of pet are you looking for?”
        - Dog
        - Cat
    - “Which gender do you prefer?”
        - Male
        - Female
        - Not important
    - “What age do you prefer?”
        - Puppy (0 - 12 months)
        - Young (13 months - 3 years)
        - Adult (4 years – 10 years)
        - Senior (more than 11 years)

- "Do you already have a pet?"
  - Yes : "Select which of these"
    - Male dog
    - Female dog
    - Male cat
    - Female cat
  - No

- "If you have a garden or a terrace, will the pet be sleeping outside?"
  - Yes
  - No

- "How much time will the pet be left alone at home during the day?"
  - 1-3 hours
  - 4-6 hours
  - More than 6 hours

- "Have you ever had a pet?"
  - No, this is my first time
  - Yes

- "Are you willing to sterilize your pet when it will be necessary?"
  - Yes
  - No

- (for dogs) "If needed, would you start a program of dog education?"
  - Yes
  - No

- "Are you willing to adopt a disabled pet?"
  - Yes
  - No

- "Are you looking for adoption in a specific range?"
  - Yes
  - No (*remember that you will pay for all the transfer costs)

- "If yes, insert your city:"

2 User Stories

- **As a user, I want to make a research of the adoptable pets based on the answers of a questionnaire, so that I can adopt a pet which is compatible with me.**
- **As a logged user, I want to save a specific pet, so that I can maintain a list of pets that I might adopt.**
- **As a logged user, I want to be notified via mail each time a pet I saved has been adopted, so that I know that I can't adopt it anymore.**

- **As a shelter manager, I want to register all the adoptable pets with their information, so that they can find a home.**
- **As a logged user, I want to make a request to meet a pet on a specific day and time, so that I can understand if it's right for me.**
- **As a shelter manager, I want to manage the meeting requests, so that I can accept / reject it or propose a different day and time which works for me.**

- As a logged user / shelter manager, I want to add to Google Calendar a confirmed appointment, so that I can have a reminder of the appointment.
- As a logged user, I want to react to the change of day and time proposed by the shelter to which I sent a meeting request, so that I can accept, propose a different day and time or annul the request.
- As a shelter manager, I want to register my shelter with its name, address, web site, telephone number, logo, so that I can show pets.
- As a shelter manager, I want to provide a wish list of products on Amazon, so that people can donate them to my pets.
- As a user, I want to receive a list of all the shelters in a city, so that I can look for dogs/cats that I might adopt.

3 Functional Requirements

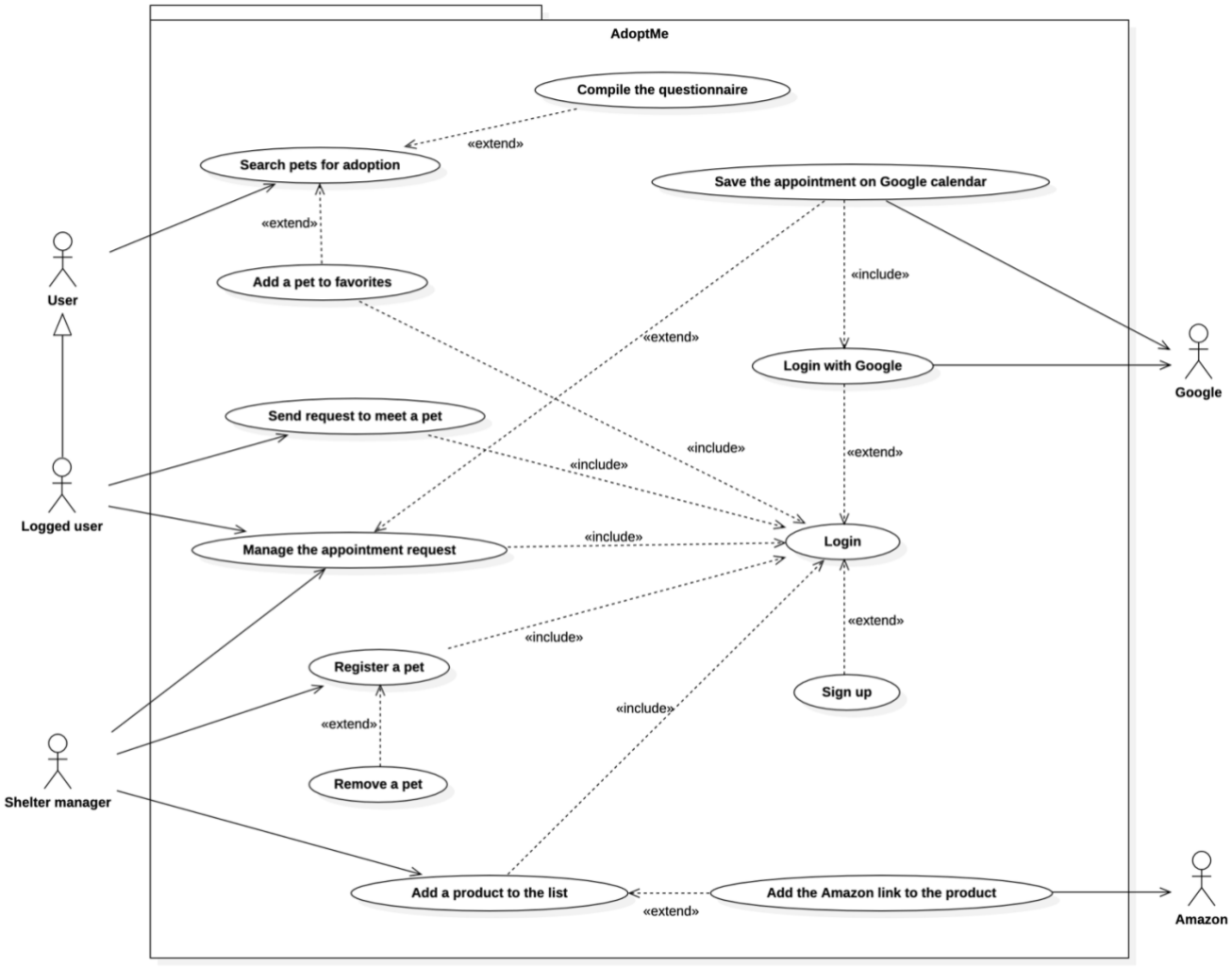
- **The system shall show a filtered list of pets, with their main information, based on data collected from the user.**
- **The system shall display a list of user's favorites pets with their main information.**
- **The system shall notify via mail the user of the adoption of a pet from their list of favorites pets with the following message: "The pet <*name of the pet*> from your list of favorites is no more adoptable."**

- **The system shall allow registration of pets with their information.**
- **The system shall notify via mail the shelter manager of the request to meet one of their pet with the following message: "The user <*name of the user*> wants to meet the pet <*name of the pet [id of the pet]* > on <*date and hour requested*>. Check on the app to accept, reject or propose a new date and/or hour."**
- **The system shall add to Google Calendar a confirmed appointment.**

- The system shall provide a list of available time slots to meet a pet.
- The system shall provide a login based on the Google account
- The system shall display lists of links to products on Amazon given by the shelter managers.

4 Use Cases

4.1 Diagram



4.2 Internal steps

Name: Register a pet

1. The shelter manager requests to register a pet
2. The system provides a blank form for the insertion of the pet's information
3. The shelter manager fills in the form with the information
4. The system provides a form to select the method to upload the photo
5. The shelter manager chooses the method

    5a. *The shelter manager chooses to take new photo*: The system opens the camera, and the shelter manager takes the photo and uploads it

6. The shelter manager selects and upload the photo from the gallery
7. The shelter manager indicates that the form is completed

    7a. *Mandatory information are missing:* The system notifies the shelter manager and sends them back to the 3rd step

    7b. *Saving process failed:* The system notifies the shelter manager and terminates the use case

8. The system validates the registration

Name: Request to meet a pet

1. The user requests to meet a pet

    1a. *The user is not logged:* The system sends them to the **login** use case

    1b. *Due to synchronization problems the pet is not adoptable anymore:* The system notifies the user and terminates the use case

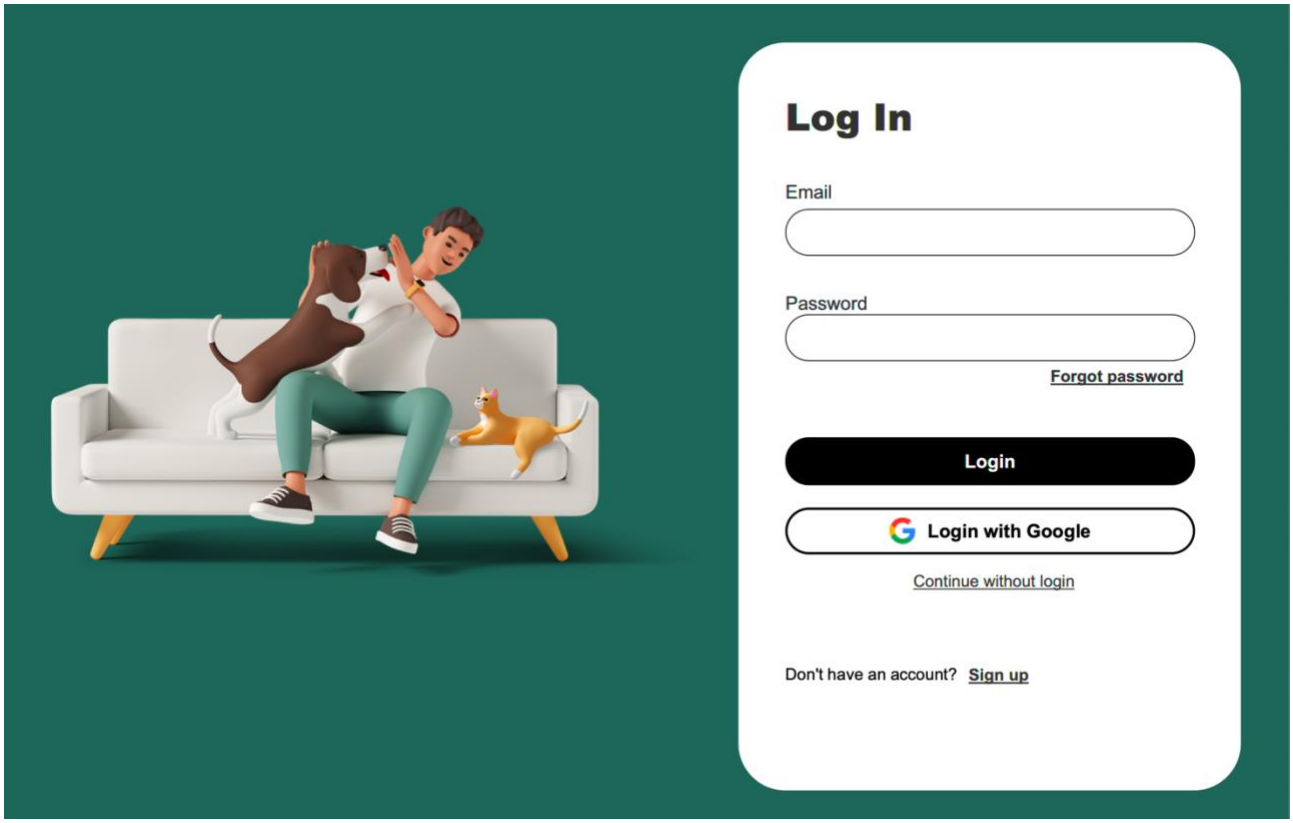
2. The system provides a form to select month, day and hour for the meeting
3. The user selects month, day and hour
4. The user indicates that the form is completed
5. The system notifies via email the manager of the shelter that holds the pet

5 Storyboards

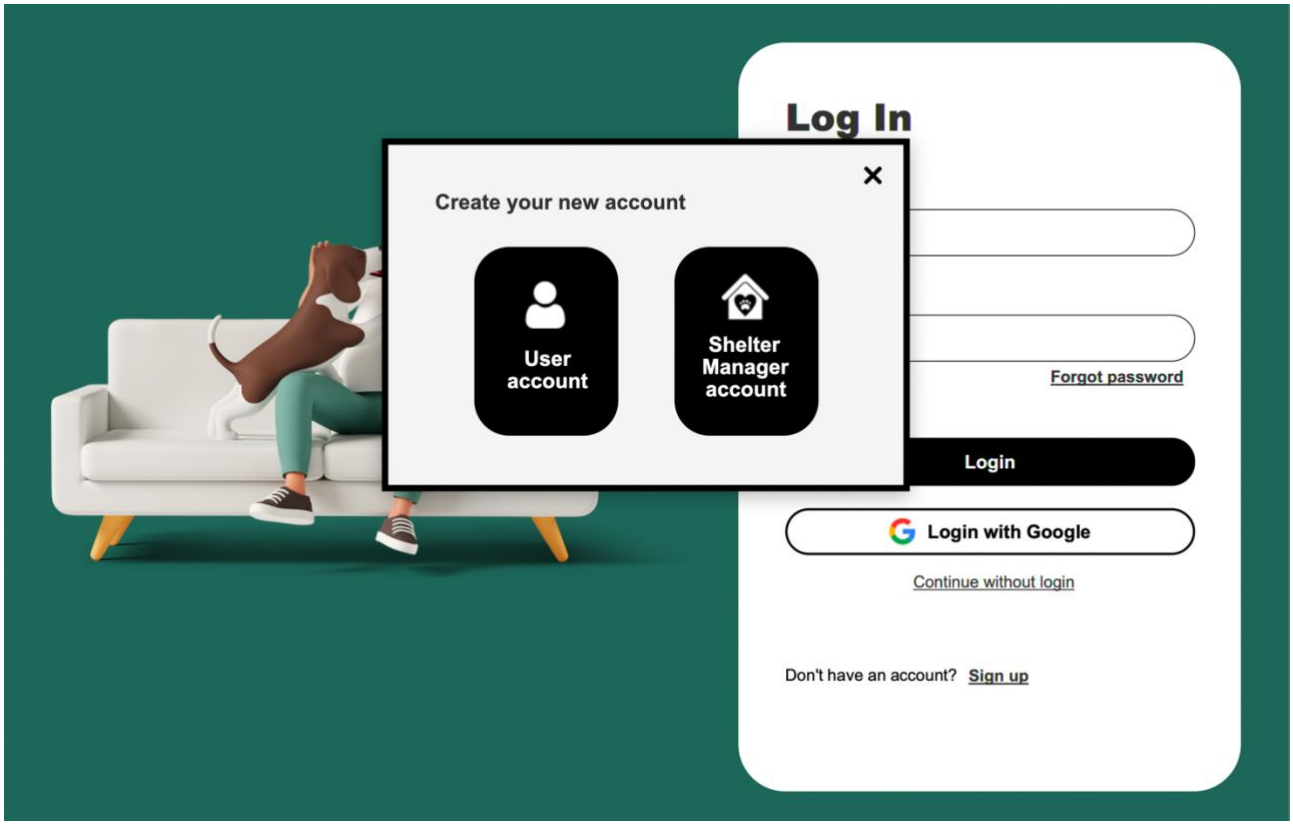
1. Login page
   1.1. Sign up page: choose account type
      1.1.1. User sign up
      1.1.2. Shelter sign up

2. Shelter homepage
   2.1. Form to register new pets
   2.2. Pet's information page
   2.3. Appointments page

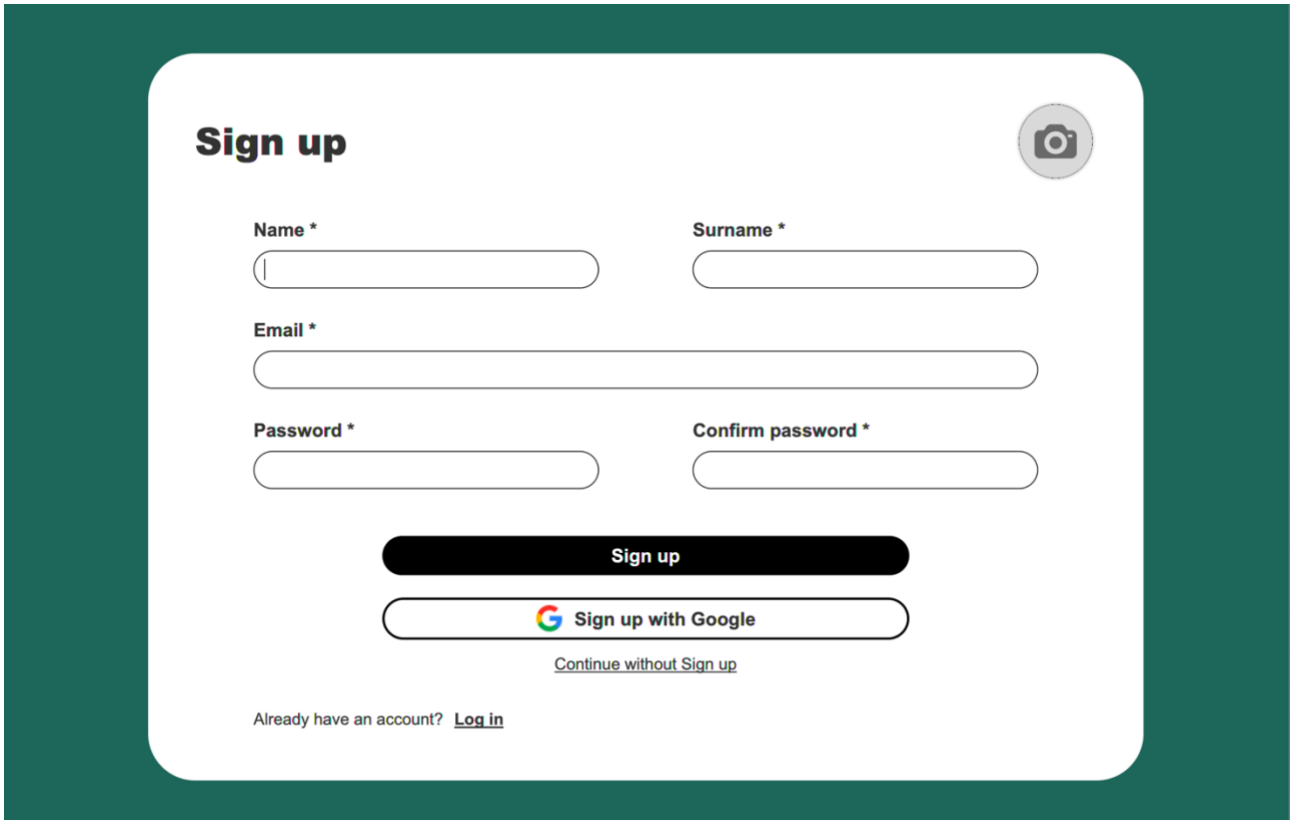
3. User homepage
   3.1. Questionnaire
   3.2. Pet's information page (it also contains the box to send a request and the button "Add to favorites")
   3.3. Appointments page



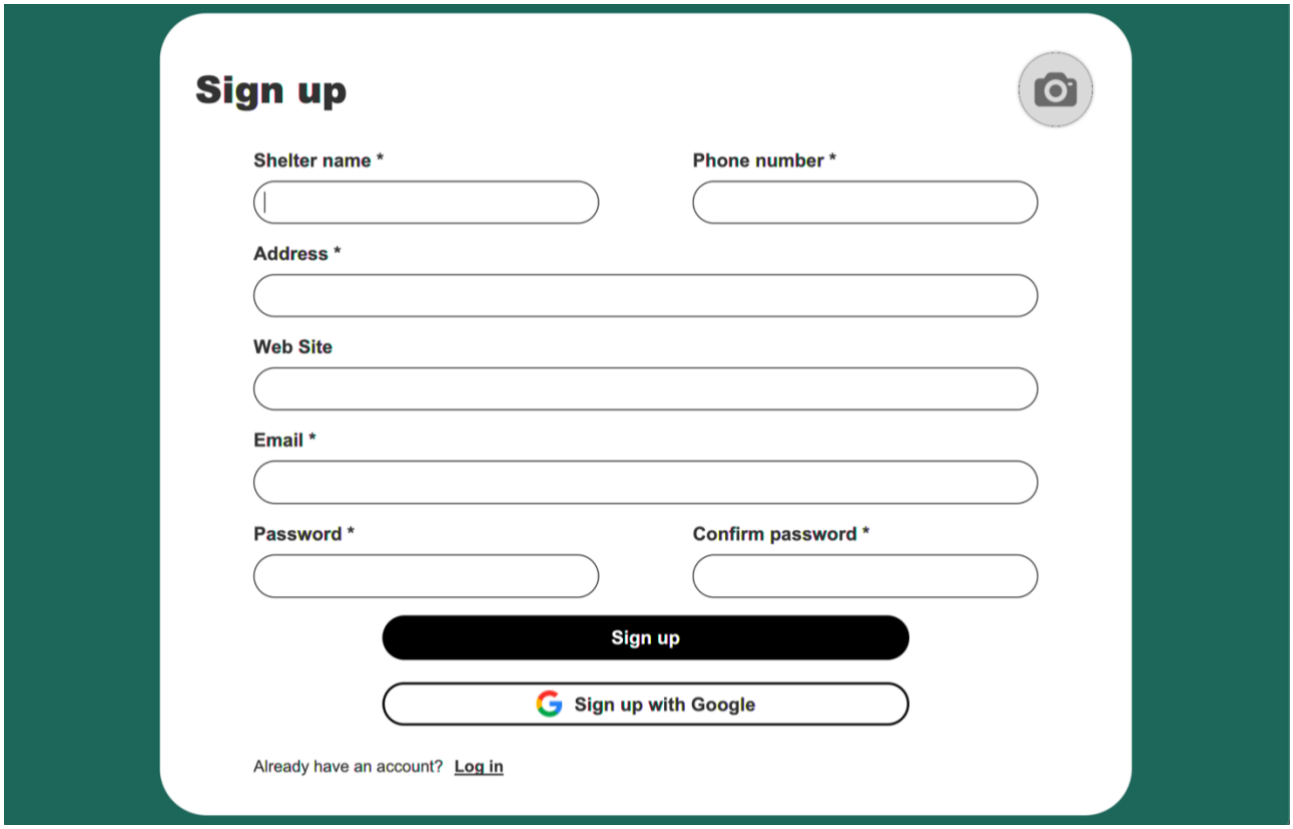
← go to index



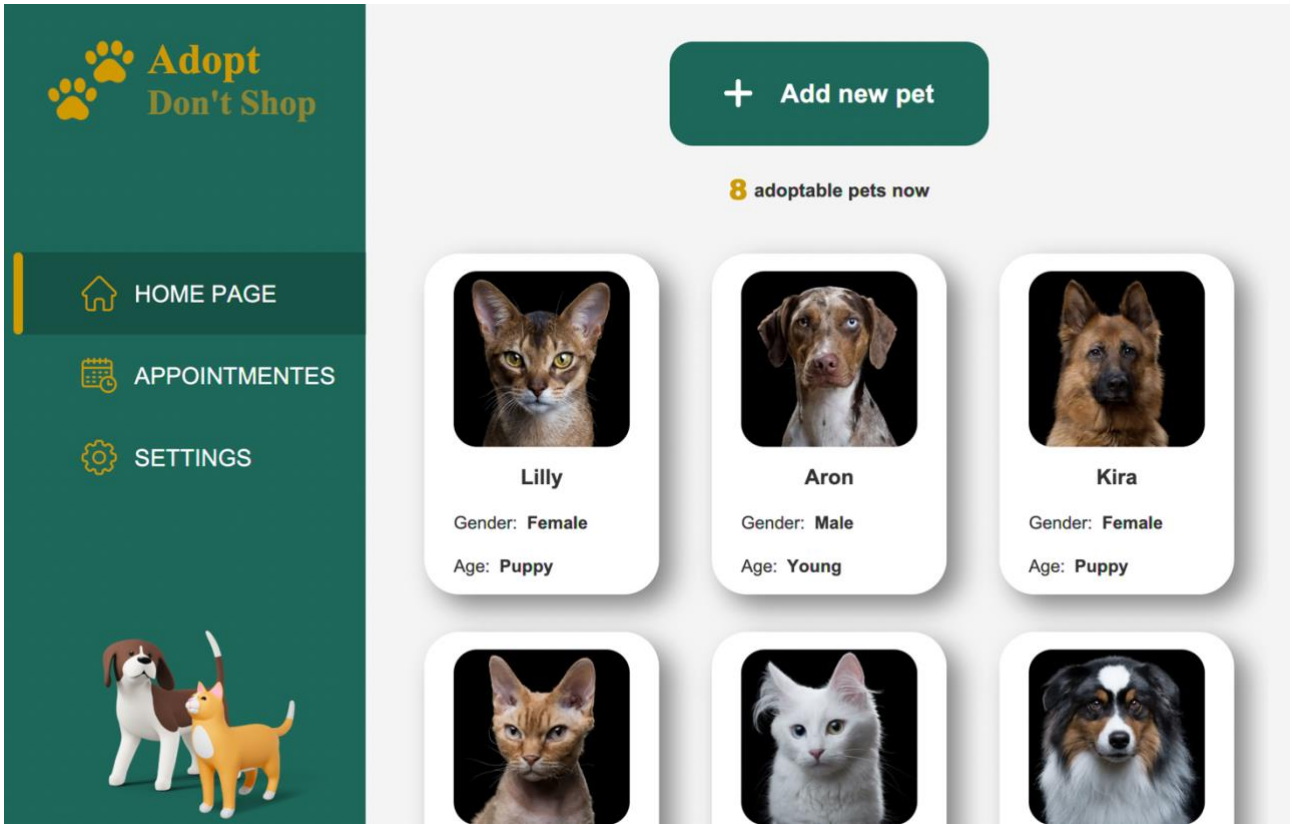
← go to index



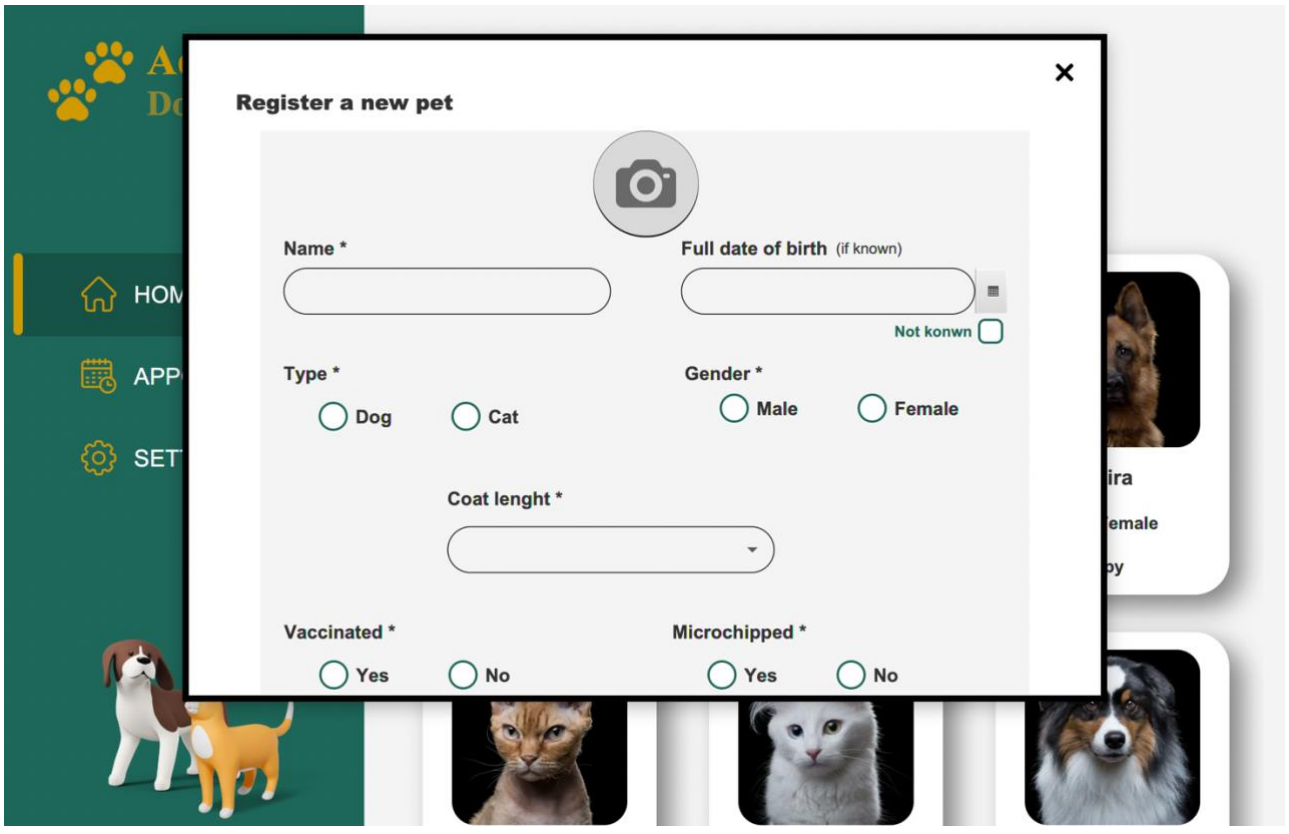
← go to index


Sign up
Shelter name *
Phone number *
Address *
Web Site
Email *
Password *
Confirm password *
Sign up
Sign up with Google
Already have an account? Log in

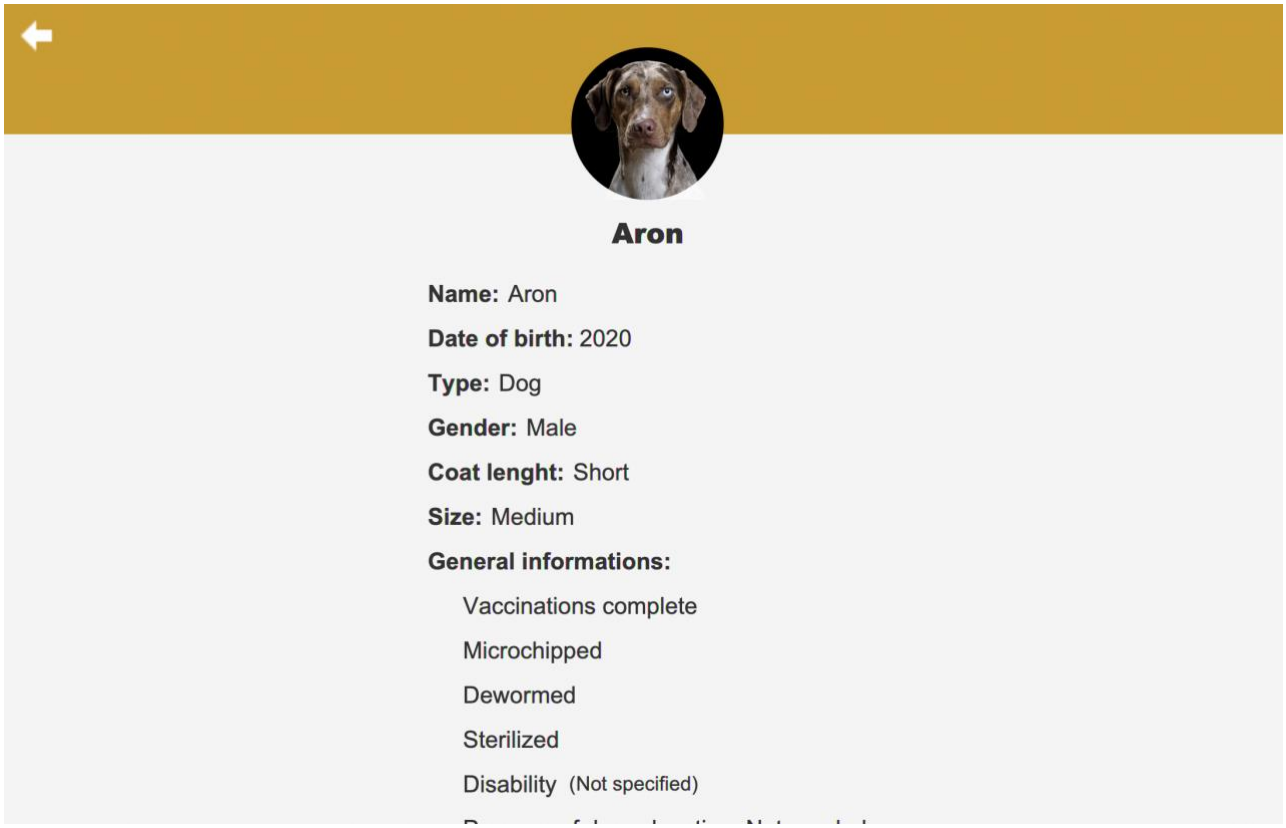
← go to index



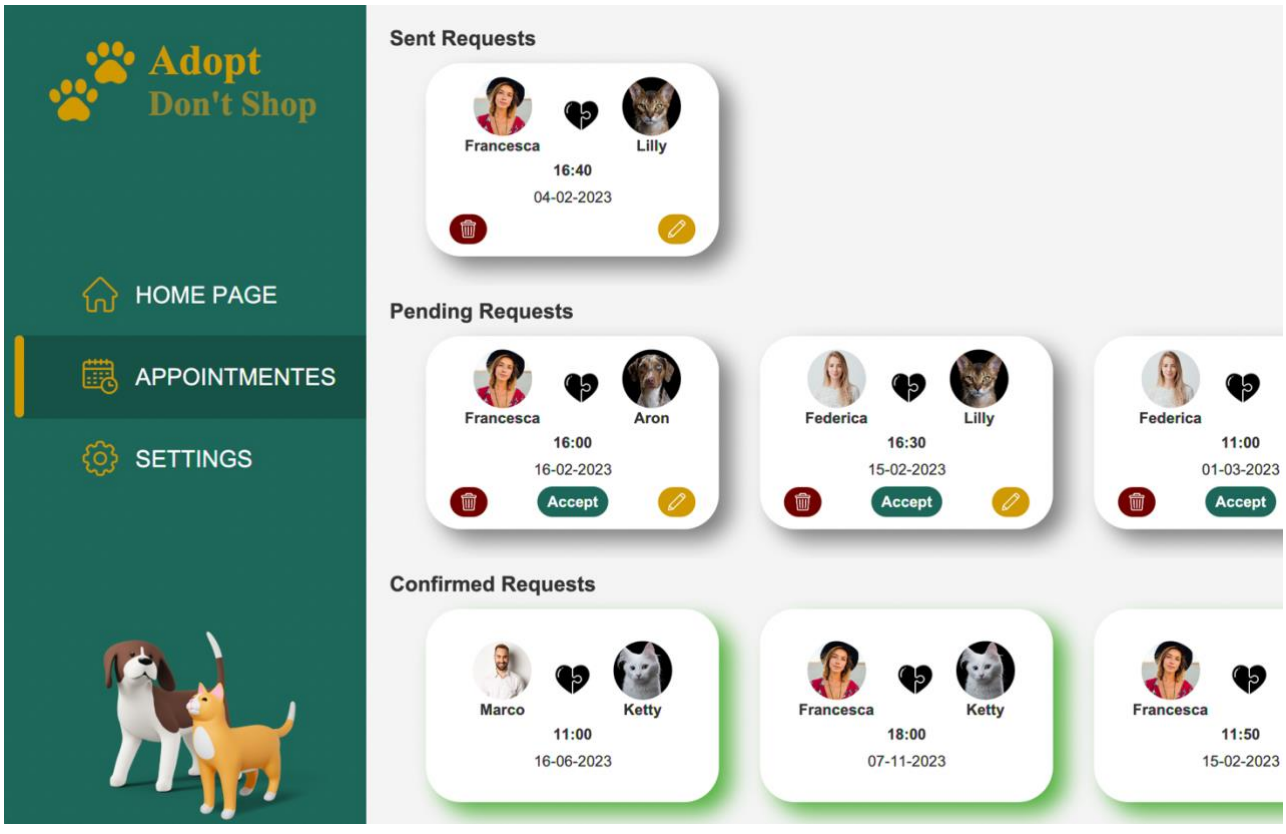
← go to index



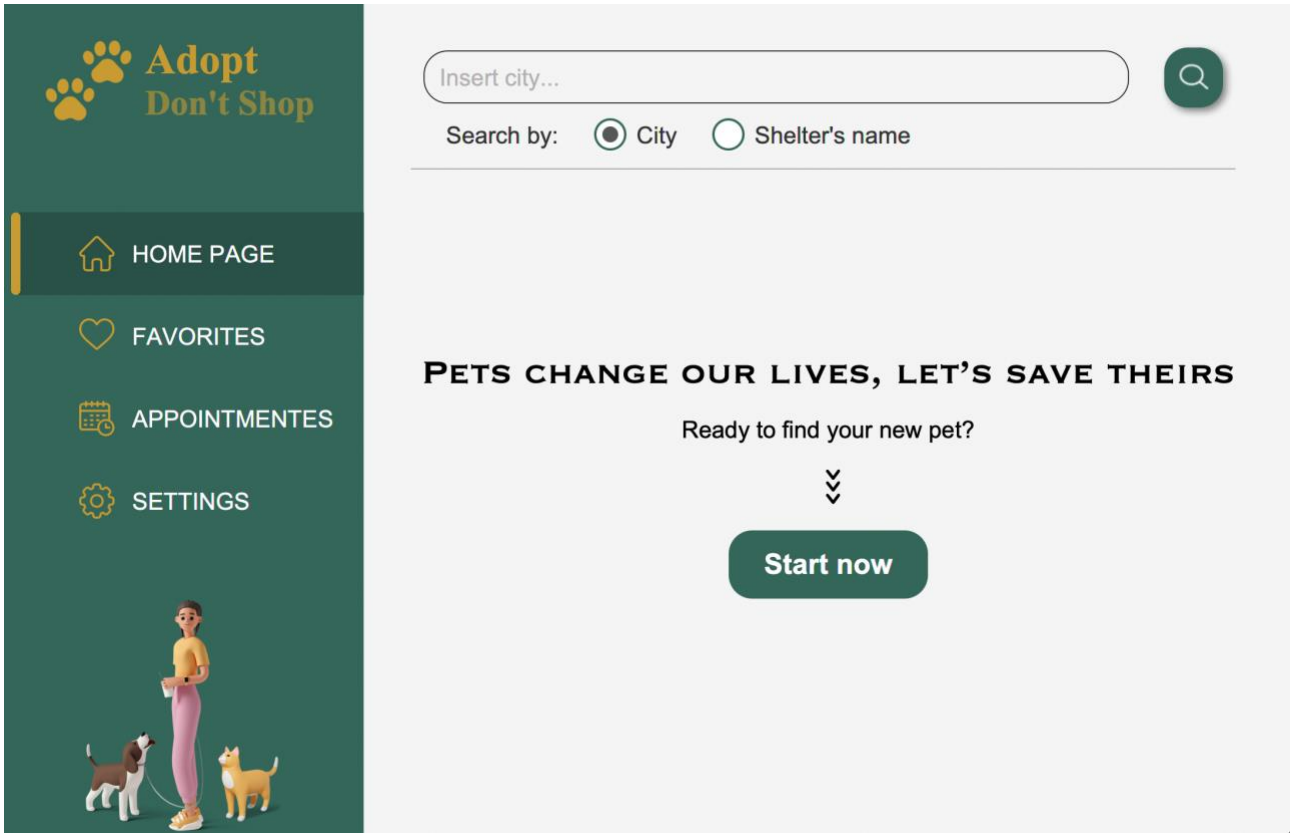
← go to index



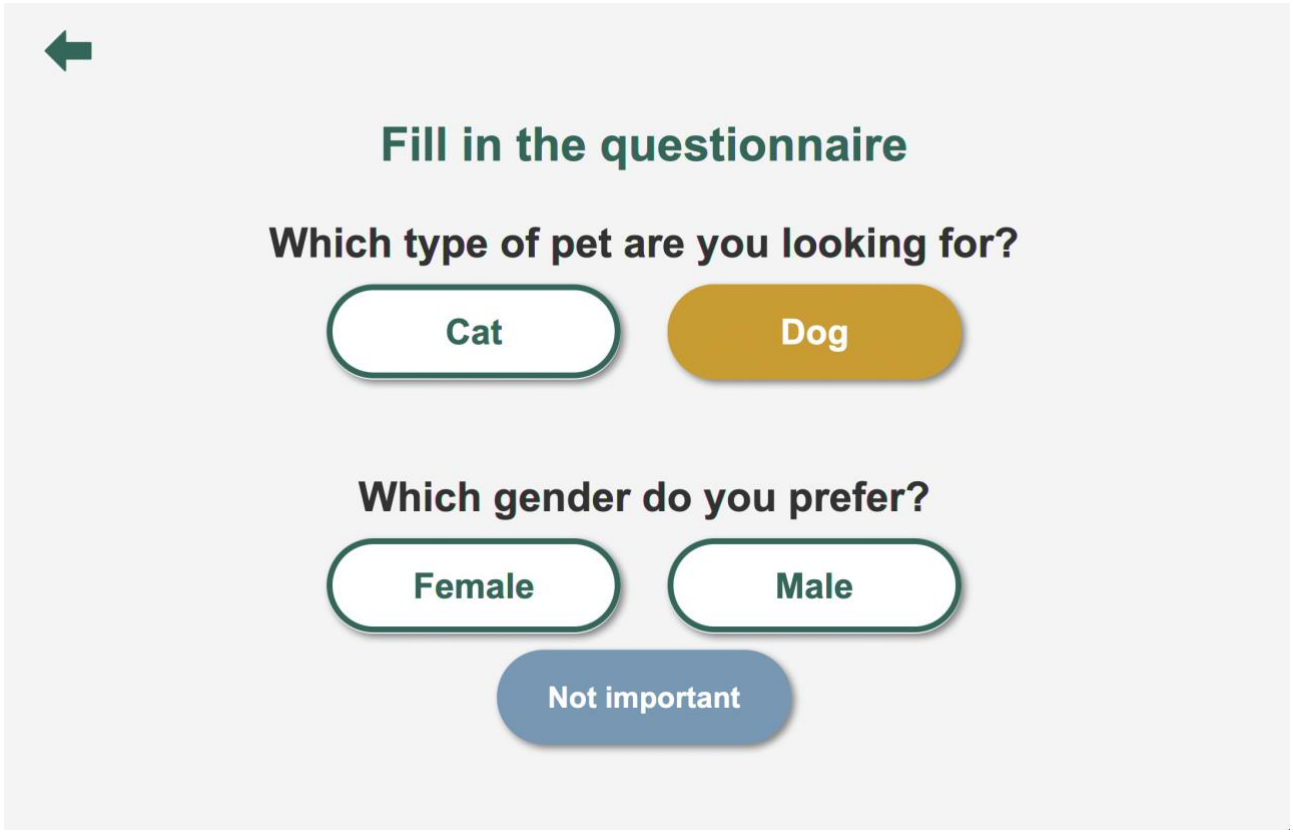
← go to index



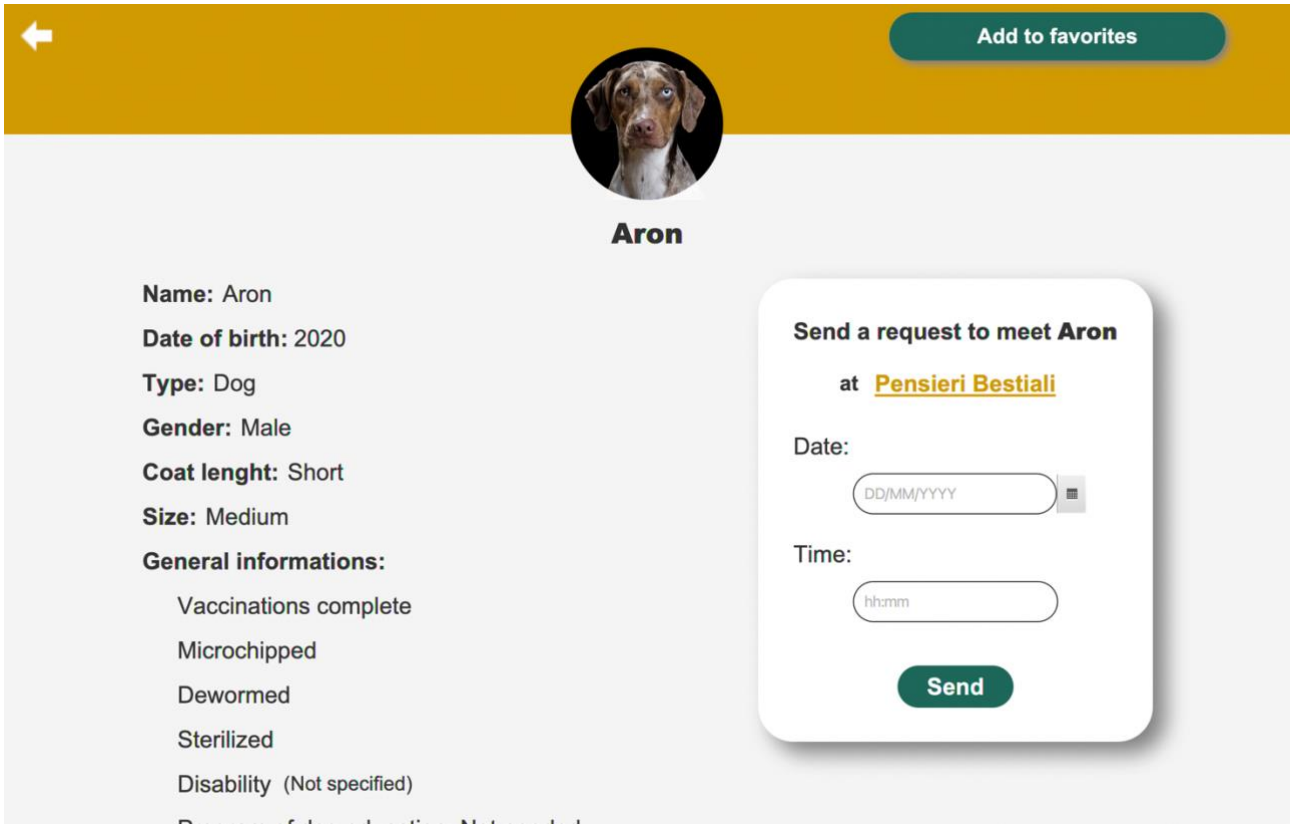
← go to index



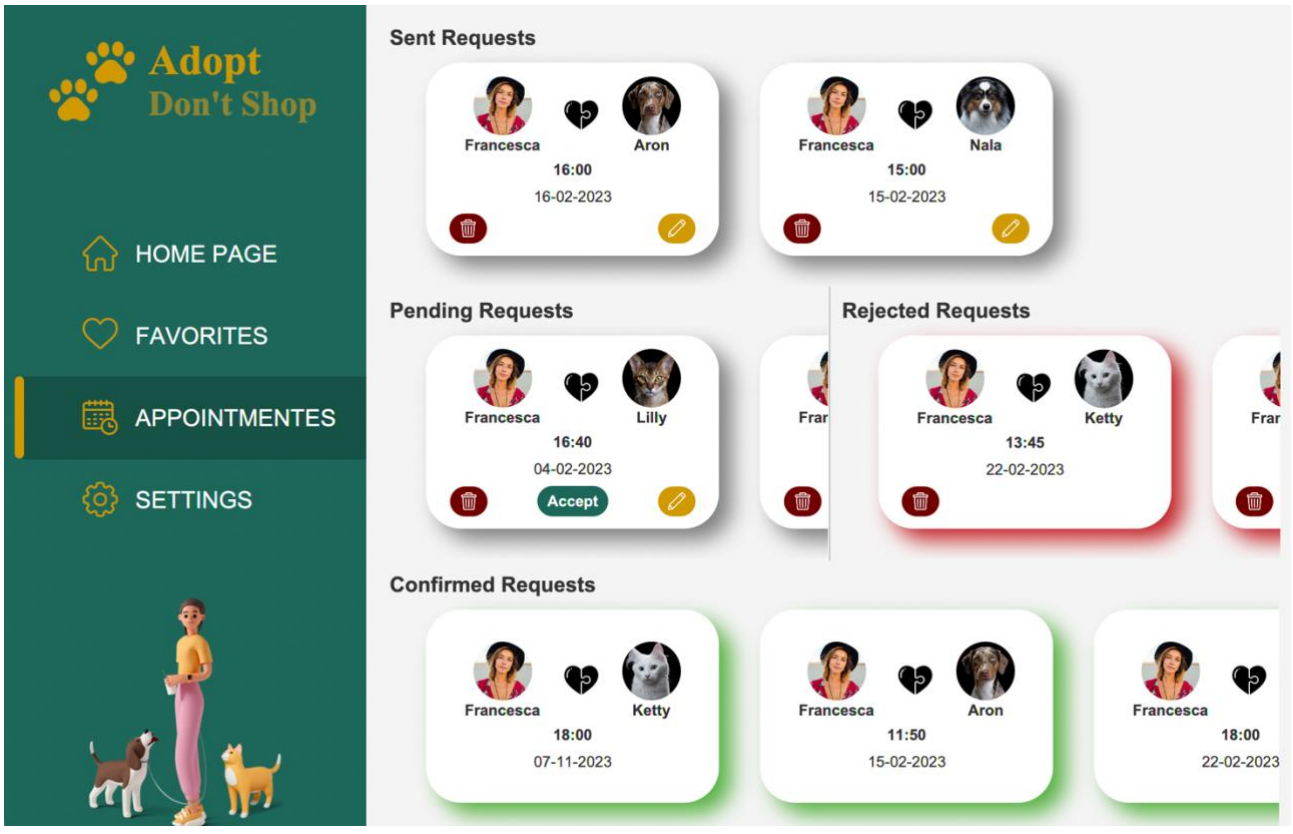
← go to index



← go to index



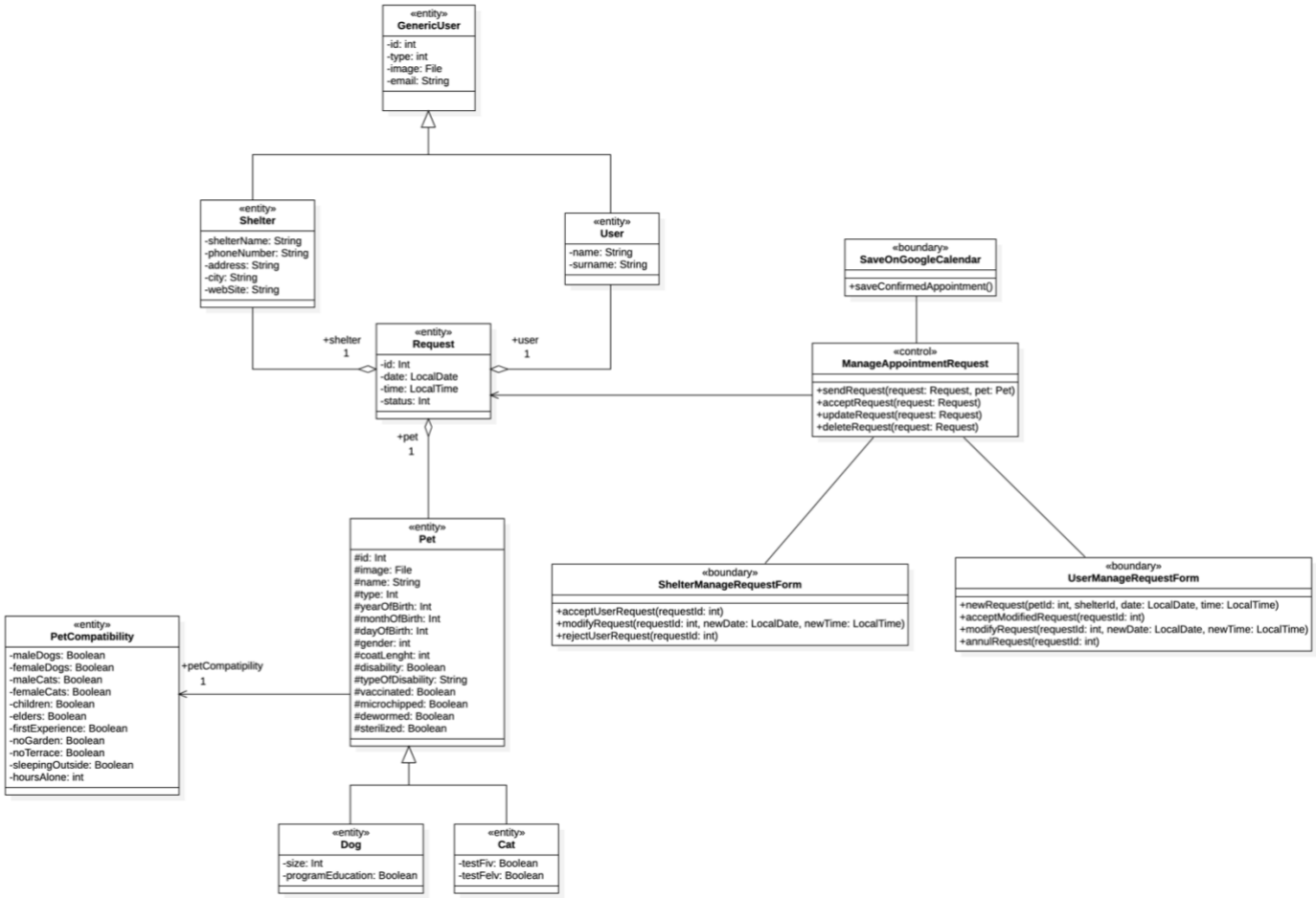
← go to index

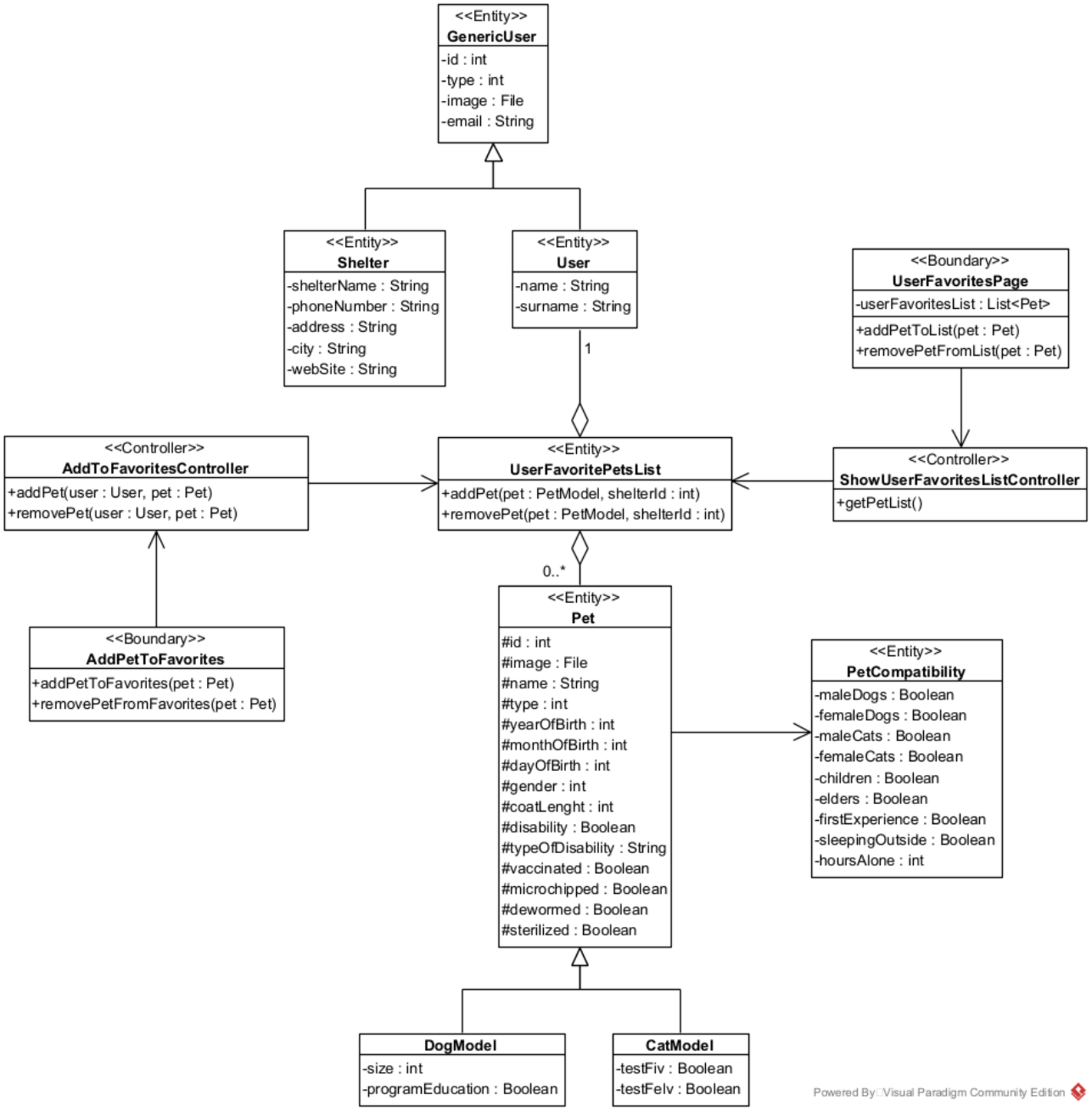


← go to index

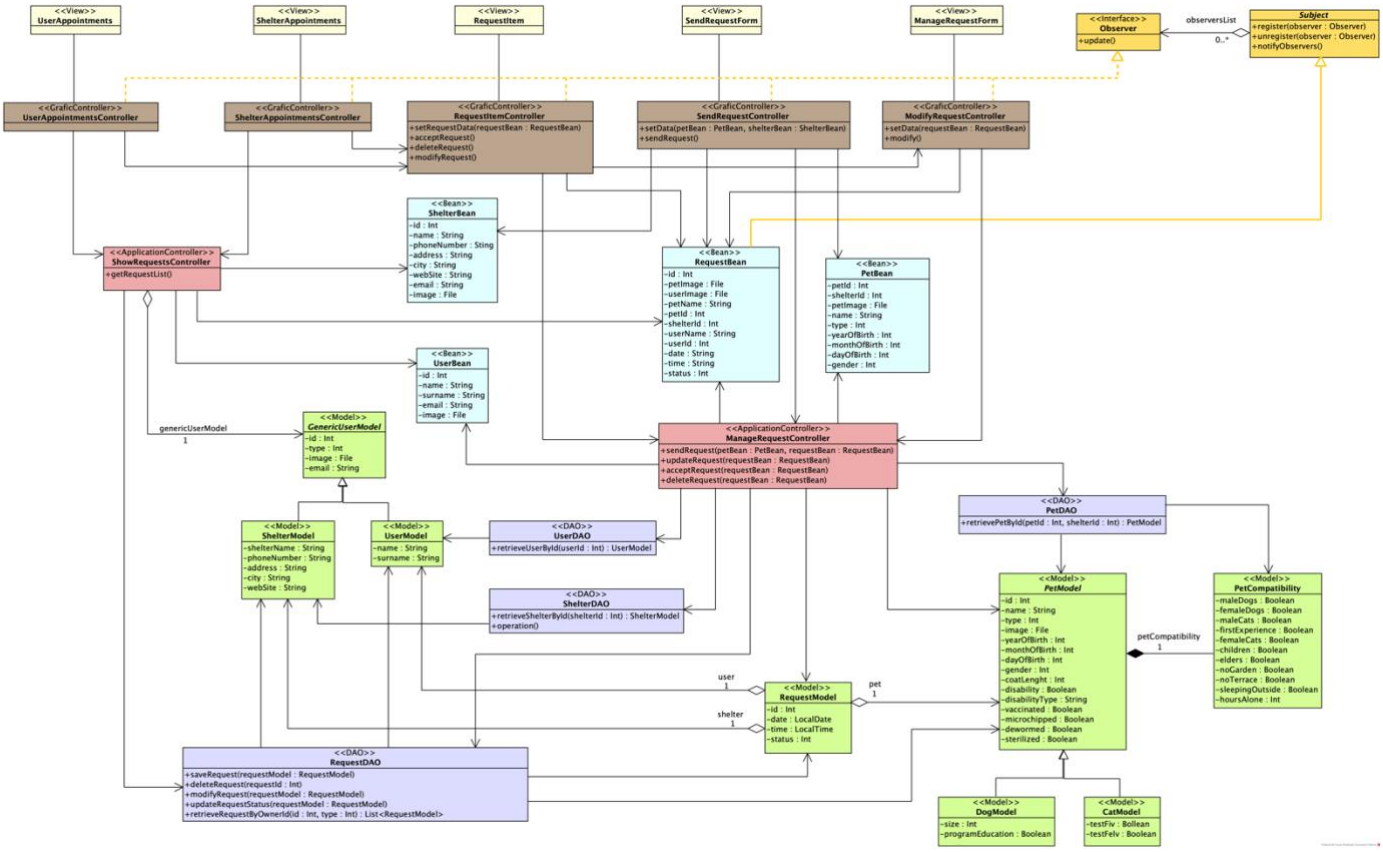
6 Class Diagram

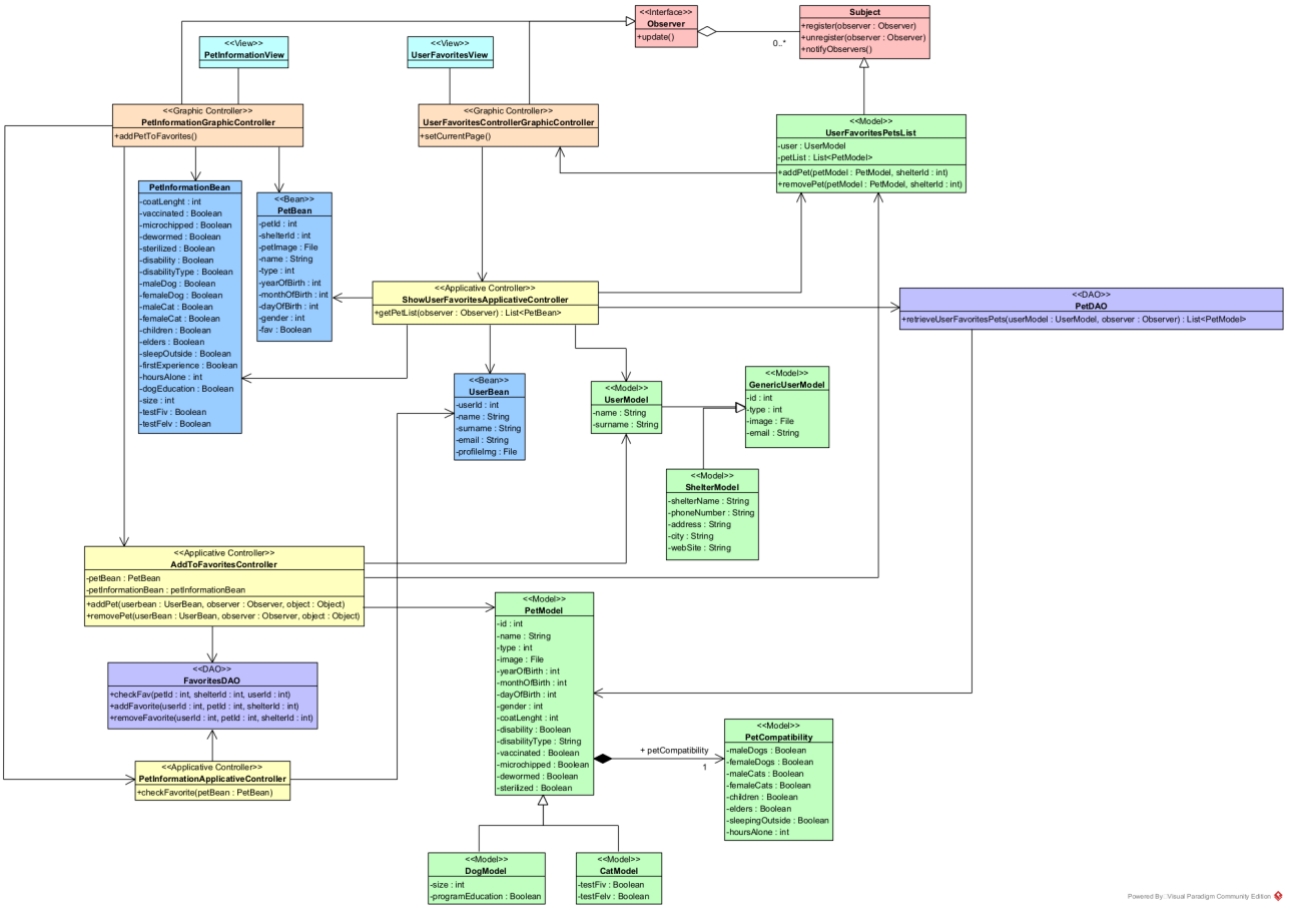
6.1 VOPC



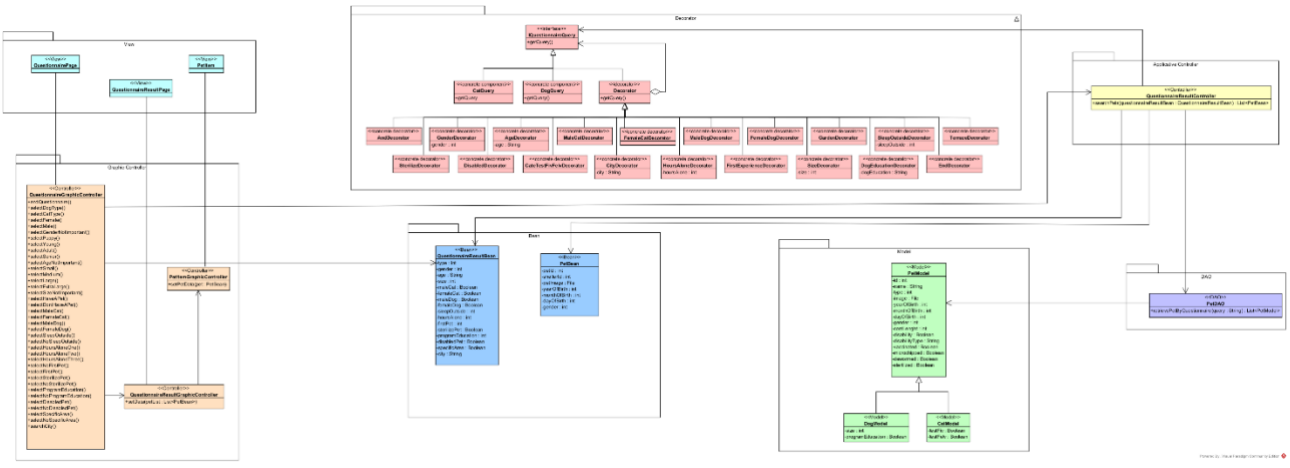

<<Entity>>
GenericUser
-id : int
-type : int
-image : File
-email : String
<<Entity>>
Shelter
-shelterName : String
-phoneNumber : String
-address : String
-city : String
-webSite : String
<<Entity>>
User
-name : String
-surname : String
1
<<Boundary>>
UserFavoritesPage
-userFavoritesList : List<Pet>
+addPetToList(pet : Pet)
+removePetFromList(pet : Pet)
<<Controller>>
AddToFavoritesController
+addPet(user : User, pet : Pet)
+removePet(user : User, pet : Pet)
<<Entity>>
UserFavoritePetsList
+addPet(pet : PetModel, shelterId : int)
+removePet(pet : PetModel, shelterId : int)
<<Controller>>
ShowUserFavoritesListController
+getPetList()
0..*
<<Boundary>>
AddPetToFavorites
+addPetToFavorites(pet : Pet)
+removePetFromFavorites(pet : Pet)
<<Entity>>
Pet
#id : int
#image : File
#name : String
#type : int
#yearOfBirth : int
#monthOfBirth : int
#dayOfBirth : int
#gender : int
#coatLenght : int
#disability : Boolean
#typeOfDisability : String
#vaccinated : Boolean
#microchipped : Boolean
#dewormed : Boolean
#sterilized : Boolean
<<Entity>>
PetCompatibility
-maleDogs : Boolean
-femaleDogs : Boolean
-maleCats : Boolean
-femaleCats : Boolean
-children : Boolean
-elders : Boolean
-firstExperience : Boolean
-sleepingOutside : Boolean
-hoursAlone : int
DogModel
-size : int
-programEducation : Boolean
CatModel
-testFiv : Boolean
-testFelv : Boolean
Powered By Visual Paradigm Community Edition

6.2 MVC

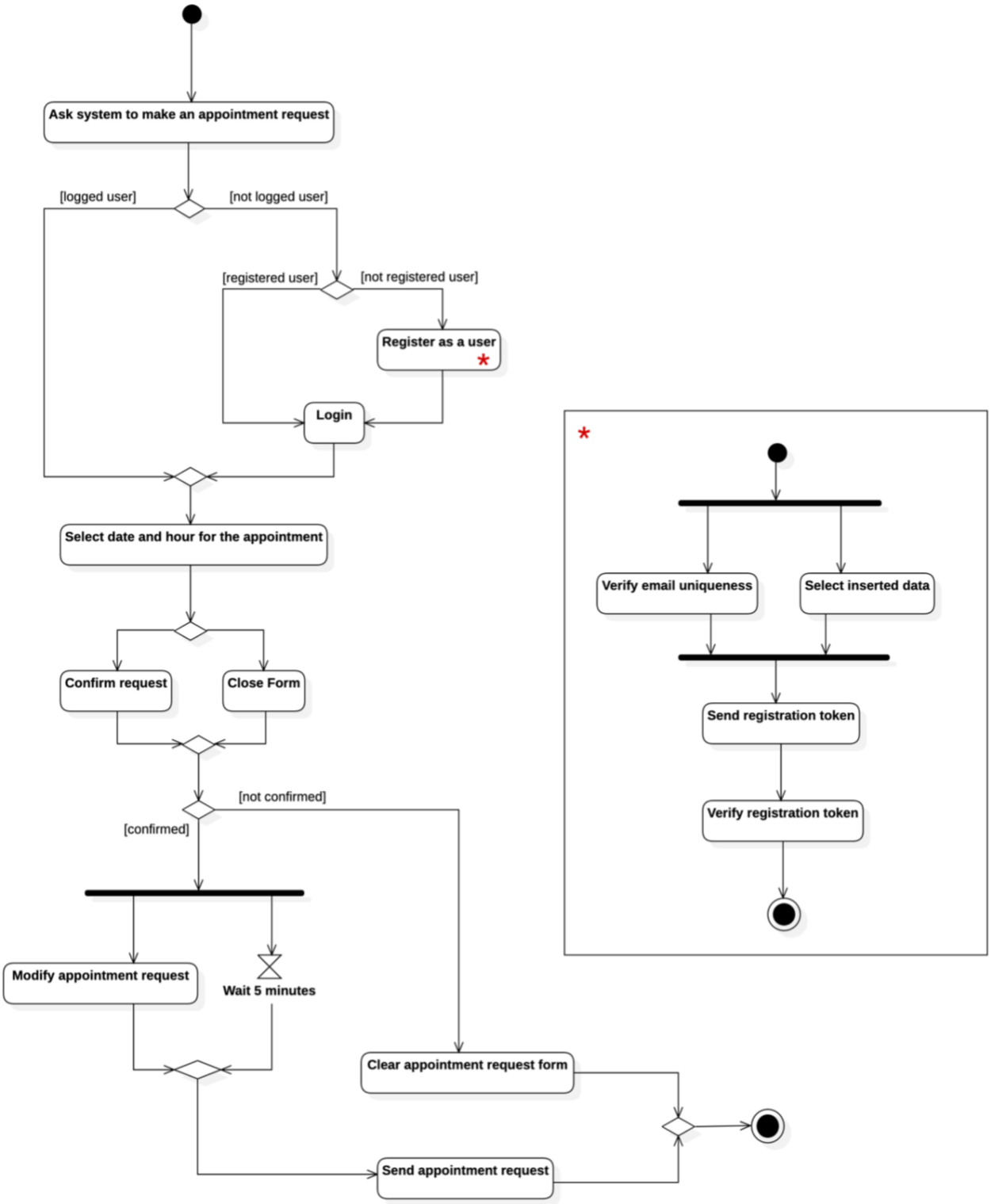

<<View>>
UserAppointments
<<View>>
ShelterAppointments
<<View>>
RequestItem
<<View>>
SendRequestForm
<<View>>
ManageRequestForm
<<Interface>>
Observer
+update()
observersList
0..*
Subject
+register(observer : Observer)
+unregister(observer : Observer)
+notifyObservers()
<<GraficController>>
UserAppointmentsController
<<GraficController>>
ShelterAppointmentsController
<<GraficController>>
RequestItemController
+setRequestData(requestBean : RequestBean)
+acceptRequest()
+deleteRequest()
+modifyRequest()
<<GraficController>>
SendRequestController
+setData(petBean : PetBean, shelterBean : ShelterBean)
+sendRequest()
<<GraficController>>
ModifyRequestController
+setData(requestBean : RequestBean)
+modify()
<<Bean>>
ShelterBean
-id : Int
-name : String
-phoneNumber : Sting
-address : String
-city : String
-webSite : String
-email : String
-image : File
<<ApplicationController>>
ShowRequestsController
+getRequestList()
<<Bean>>
RequestBean
-id : Int
-petImage : File
-userImage : File
-petName : String
-petId : Int
-shelterId : Int
-userName : String
-userId : Int
-date : String
-time : String
-status : Int
<<Bean>>
PetBean
-petId : Int
-shelterId : Int
-petImage : File
-name : String
-type : Int
-yearOfBirth : Int
-monthOfBirth : Int
-dayOfBirth : Int
-gender : Int
<<Bean>>
UserBean
-id : Int
-name : String
-surname : String
-email : String
-image : File
genericUserModel
1
<<Model>>
GenericUserModel
-id : Int
-type : Int
-image : File
-email : String
<<ApplicationController>>
ManageRequestController
+sendRequest(petBean : PetBean, requestBean : RequestBean)
+updateRequest(requestBean : RequestBean)
+acceptRequest(requestBean : RequestBean)
+deleteRequest(requestBean : RequestBean)
<<DAO>>
PetDAO
+retrievePetById(petId : Int, shelterId : Int) : PetModel
<<Model>>
ShelterModel
-shelterName : String
-phoneNumber : String
-address : String
-city : String
-webSite : String
<<Model>>
UserModel
-name : String
-surname : String
<<DAO>>
UserDAO
+retrieveUserById(userId : Int) : UserModel
<<DAO>>
ShelterDAO
+retrieveShelterById(shelterId : Int) : ShelterModel
+operation()
<<Model>>
PetModel
-id : Int
-name : String
-type : Int
-image : File
-yearOfBirth : Int
-monthOfBirth : Int
-dayOfBirth : Int
-gender : Int
-coatLenght : Int
-disability : Boolean
-disabilityType : String
-vaccinated : Boolean
-microchipped : Boolean
-dewormed : Boolean
-sterilized : Boolean
<<Model>>
PetCompatibility
-maleDogs : Boolean
-femaleDogs : Boolean
-maleCats : Boolean
-firstExperience : Boolean
-femaleCats : Boolean
-children : Boolean
-elders : Boolean
-noGarden : Boolean
-noTerrace : Boolean
-sleepingOutside : Boolean
-hoursAlone : Int
petCompatibility
1
user
1
shelter
1
pet
1
<<Model>>
RequestModel
-id : Int
-date : LocalDate
-time : LocalTime
-status : Int
<<DAO>>
RequestDAO
+saveRequest(requestModel : RequestModel)
+deleteRequest(requestId : Int)
+modifyRequest(requestModel : RequestModel)
+updateRequestStatus(requestModel : RequestModel)
+retrieveRequestByOwnerId(id : Int, type : Int) : List<RequestModel>
<<Model>>
DogModel
-size : Int
-programEducation : Boolean
<<Model>>
CatModel
-testFiv : Boolean
-testFelv : Boolean

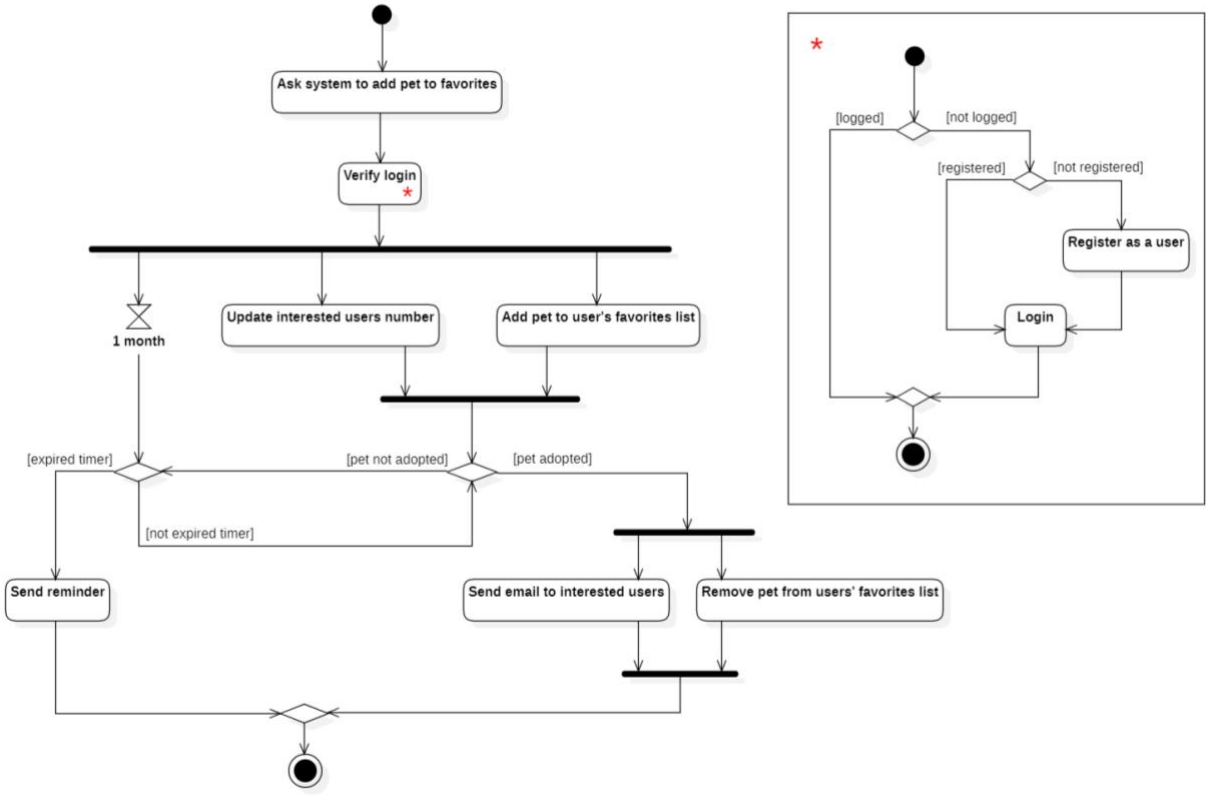


6.3 Design Pattern

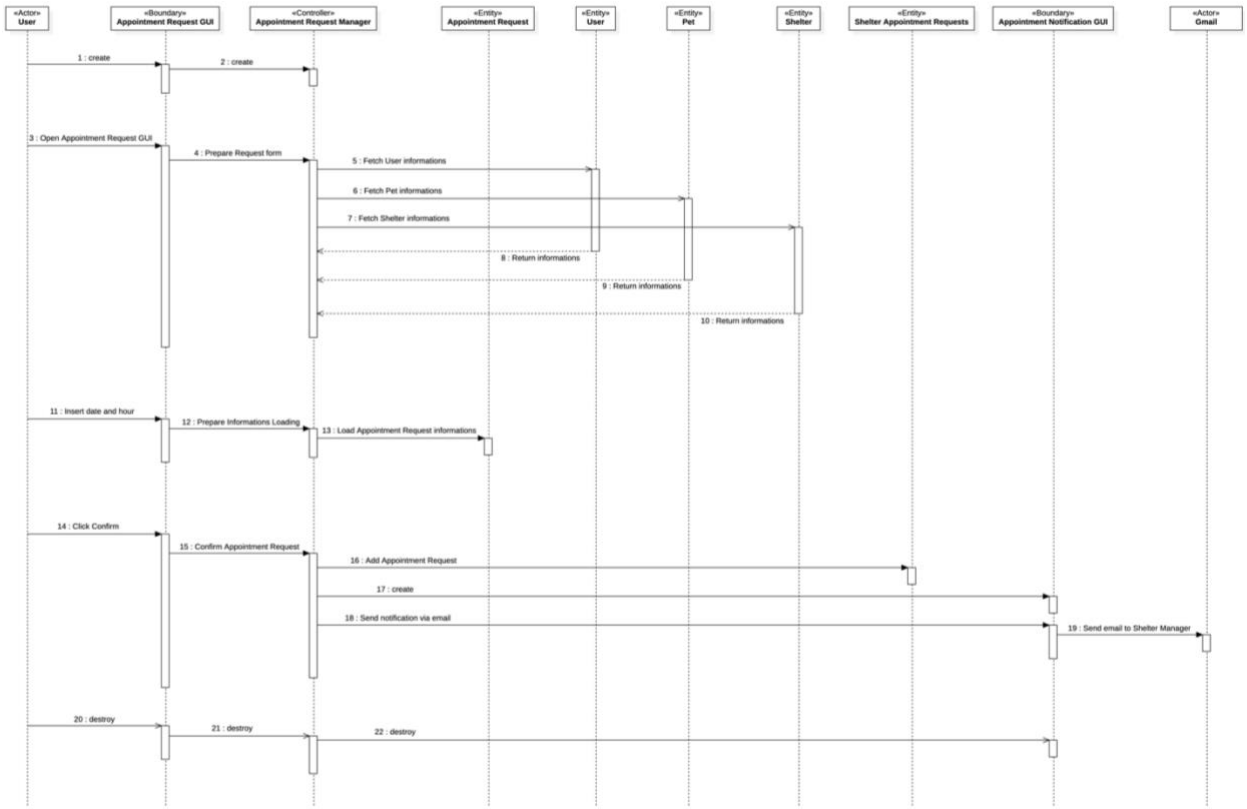


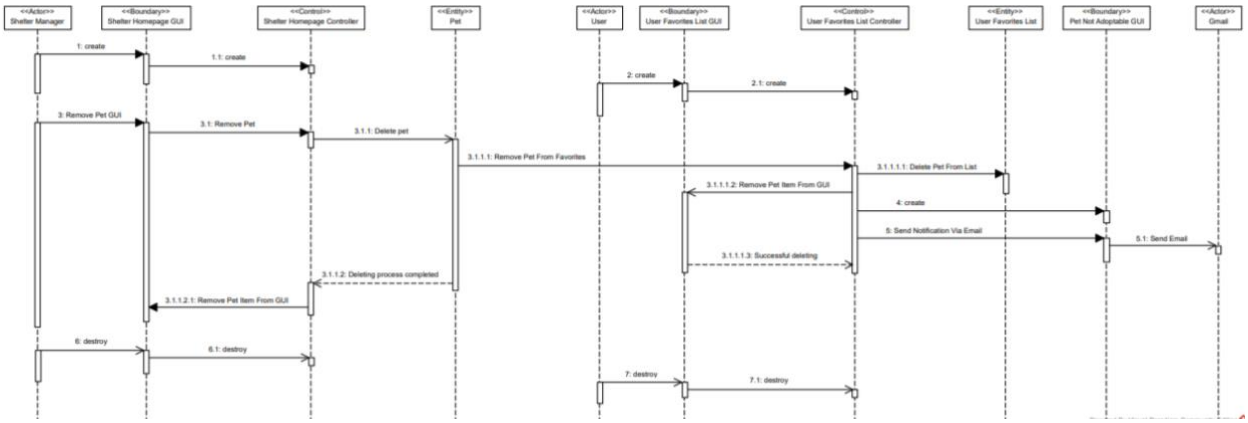
7 Activity Diagram



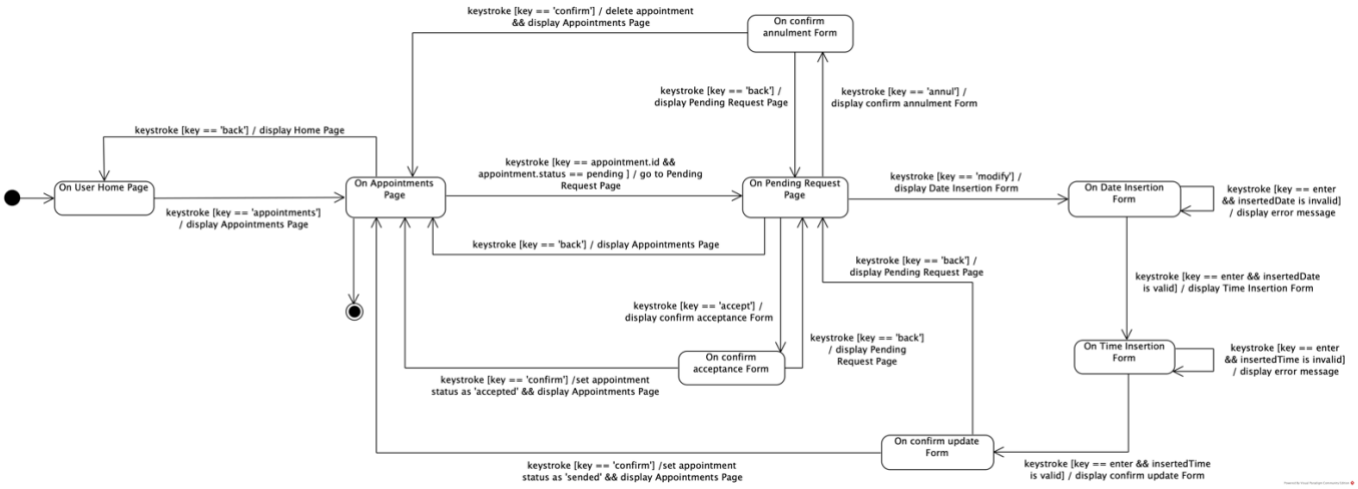

Ask system to add pet to favorites
Verify login
*
1 month
Update interested users number
Add pet to user's favorites list
[expired timer]
[pet not adopted]
[pet adopted]
[not expired timer]
Send reminder
Send email to interested users
Remove pet from users' favorites list
*
[logged]
[not logged]
[registered]
[not registered]
Register as a user
Login

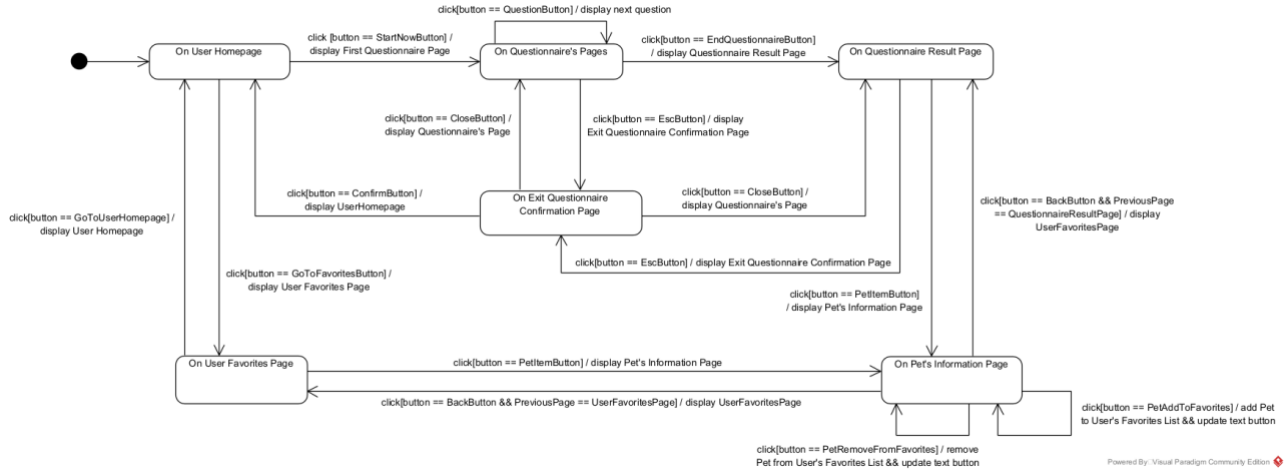
8 Sequence diagram





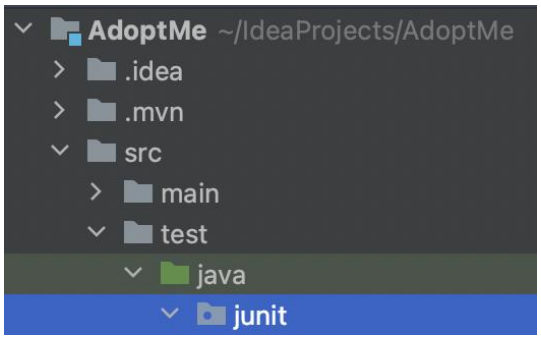
9 State diagram





10 Testing

This is the tests' package of our project



Link project's Sonar Cloud: https://sonarcloud.io/summary/overall?id=Francesca-Pavone_AdoptMe

Link project's repository: https://github.com/Francesca-Pavone/AdoptMe.git

Link deliverables' repository: https://github.com/Francesca-Pavone/AdoptMe-Deliverables.git