UNIVERSIDAD DE PUERTO RICO

RECINTO UNIVERSITARIO DE MAYAGÜEZ

Group Project:

Proposal

2Bird1Stone Members:

Efrain Oliveras **Angel Ortiz** **Victor M. Baez**

Arnaldo Villarrubia **Luis D. González** **Zulmarie Jiménez**

INSO4115 & INSO4117

Prof. Marko Schutz

February 11th, 2020

Table of Content

Informative

Domain Description

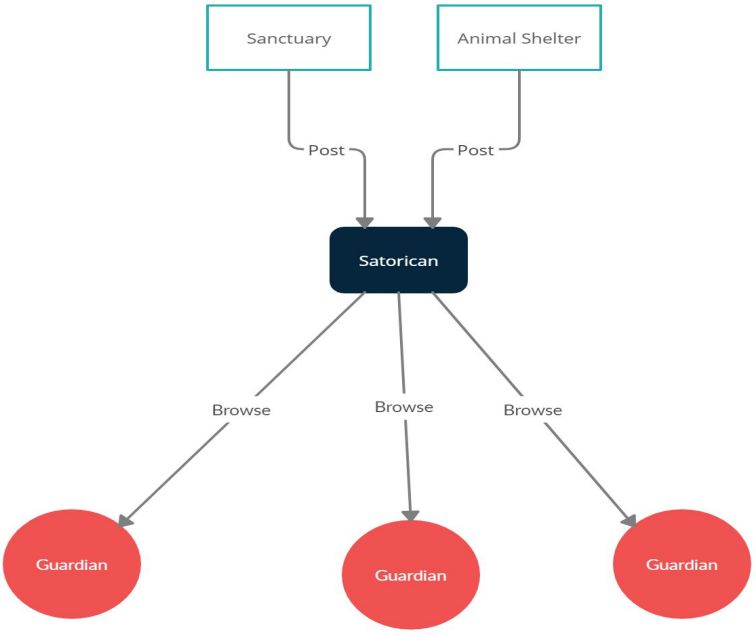
Due to the short range of coverage animal shelters have at their disposal there are alot of animals without a proper home. Since most shelters don't have enough resources to maintain a substantial amount of animals, they are forced to euthanize the sick, old and least probable to be adopted to avoid overpopulation. If the institution does not believe in euthanasia than once their facilities are at capacity they are no longer able to rescue more animals. Regardless if the institution euthanizes its animals or not costs to operate and maintain these institutions running is extremely high. Especially when its 100% financed without governmental support. Therefore the quickest solution for these institutions is for people to adopt animals.

Requirements Prescription

There is a clear need to make a system that eases the adoption process of an animal from a shelter. Few systems exist that try to tackle this issue which is why our systems would need to take this into account when developing its UI and Services. There is therefore a need for us to identify the domain for our platform and what requirements we would need to satisfy it. We would also need a proper software design and implementation so that the systems we create can have a long period of use and work as intended.

Software Design

For better grasping how the design of this application will work, the group decided to present the interaction of the two types of main users: the Sanctuary/Animal Shelter and the person looking to adopt an animal (guardian). Look at the following figure. At the top are the users responsible for adding the available animals in their facilities. This includes adding all information that is important for the guardian to analyze so he/she can make the decision to adopt or not the animal (this information is explained in the following sections). This is *posted* onto the Satorican application where the guardians can *browse* all the available options for adopting an animal. Satorican will work as a mediator between these two users.



Personas

To adequately measure our target client, our group will have to create a persona. This persona is our current goal as to the needs and characteristics of an everyday client for our application. The group will base most of its primary functions and user interface on the needs this developed persona will have. It is important to note that since this is the start of the project, the following persona description is not written in stone, meaning that it can be changed in the future as we progress with the project.

Our project's personas are two: the animal shelter and the guardian (user) willing to adopt an animal. The animal shelter persona is described as someone who has too many animals in their institution. They desperately need a way for more people to adopt their animals. If not adopted in a certain time, due to overpopulation in the institution, some animals are euthanized. The animal shelter persona must also make sure that the new caretaker of the animal does not abuse it nor abandon it, otherwise, what is the point of having more people adopt animals.

The guardian persona is looking for a way to find the perfect animal for the family. This person does not want to have to go to all the shelters available in his city (or the ones very far away) just so when he arrives he may be disappointed that the animal they wanted is not there. The guardian wants to find a way that he/she can be certain of the type of animal he/she will buy/adopt will be available in the shelter. What this person needs is a centralized way to look for the animal he/she wants, compare all the shelters that have it, and adopt the animal with the least amount of problems.

Descriptive

Rough Domain Sketch

The logistics behind Animal Rescue Assistance has a structure of entities, functions, events and behaviours. These include: the acceptance of users into the system. The browsing, searching and requesting for adoption of different animals located in multiple institutions.The acceptance or denial of the user's request for adoption. The orientation and implementation of said system for institutions (a.k.a shelters).Through communication with the institutions modifications may occur (pivots) but this will be the basis for the project. The retrieval and insertion of animal's information into the system. Available communication channels between institution and user. Be it through the system or offering information to each other. Publicly available information regarding institutions readily available in our system.

Descriptive Domain Terminology

- Cloud Services -interface incharge of notifications, creating chats, requests, etc.
- Real Time Pictures/Videos - Media (photos and videos) taken directly from the applications instead of the device's camera.
- Administrator - Developers who work to provide services for the shelters/sanctuaries. They provide maintenance, receive feedback, receive reports among other actions.
- Catalog - page where all participating institutions display their animals for adoption.
- Browsing the catalog - the action of searching without a specific goal in mind. This activity is realized by the user.
- Shelter/Institution- facilities and infrastructure incharge of housing animals to be adopted. Also known as sanctuaries, rescues, pounds, etc. This is standardized due to the moral and philosophical perspectives from each type of animal adoption organization.

Descriptive Domain Narrative

Our domain is Animal Rescue Assistance. We will rely on a few basic notions: rescued animals, authorized users and institutions among others. By Animal Rescue Assistance we refer to actions taken to develop a system for animals who have been abandoned to be rescued. A rescued animal can be adopted, euthanized, stored in a facility or moved from one facility to another.

(i) Adopting a rescued animal is when a person takes the animal from a storage facility and brings them to their house. For our project said individuals must be an authorized user.

(ii) Euthanasia is the practice of intentionally ending a life to relieve pain and suffering. Under the domain of rescued animals this practice is sometimes used when the animal has spent too much time in a facility and has not been adopted.

(iii) A rescued animal is stored in a facility in wait of being adopted. If too much time has passed in said facility and the institution does not believe in euthanasia the rescued animal will be moved to another facility to increase its odds of adoption.

(iv) Rescued animals under our project can only be adopted by authorized users. By authorized we refer to the verification process realized to confirm identity, check for a criminal record and proof of address.

(v) To confirm identity a government issued identification shall be shown via a real time picture. A valid governmentally issued identification with photo would be a passport or driver’s licence.

(vi) Proof of address is a receipt for water or electrical services provided (in the case of Puerto Rico) by the government. In this document your address shall appear. If the utility bill is under someone else’s name then a picture of a written letter with the signature of that person and an ID shall be provided in addition to the receipt.

(vii) In Puerto Rico to verify if an individual has a prior criminal record it is asked that the individual present a Certificate of Good Behavior. This is a digital certificate that the user can obtain online through the Puerto Rican Police Departments website. This certificate verifies that the user does not have a criminal record.

(viii)The difference between an animal sanctuary and an animal shelter is that sanctuaries are institutions whom do not believe in euthanasia unless recommended by the veterinarian (for medical reasons). Shelters are places who hold animals for a certain period of time in hopes of the animal being adopted. If no one decides to adopt then the animal is euthanized.

Domain Entities

1. Animal: These are the items that will be under a User or Shelter ownership.
    - Composed with Atomic Entities such as: Dogs, cats, guinea pigs, etc.
    - Each Animal has attributes: species, name, sex, age, weight, medical conditions, etc
2. User: These are people using the Satorican app with their own unique profile account.
    - Composed with Atomic Entities such as:
        - Customers: a User that only utilizes the system to adopt and take care of a certain amount of animals.
        - Staff Members: a User that utilizes the system to serve an institution. They evaluate adoption requests, take care and administrate stray animals.
    - Each User has attributes: name, age, amount of pets, occupance, residence, ect
3. Shelter: These are the buildings containing an amount of animals that are available for adoption.
    - Composed with Atomic Entities such as:
        - Staff Members: a User that utilizes the system to serve an institution. They evaluate adoption requests, take care and administrate stray animals.
        - Buildings: a location that will contain an amount of domestic animals ready to be adopted by the system (aka *Satorican*) Users.
    - Each Shelter has attributes: name, location, amount of animals, Working hours, ect
4. Administrator: These are the programmers and founders that maintain updates on the system.
    - Composed with Atomic Entities such as:
        - Programmers: Group of people working developing it and serving maintenance.
        - Managers: Group of people who work with client relationships and business inquiries.

5. Database: This will be the server containing all sensitive information about Users, Shelters and Animals. Only administrators have full access.
    - Composed with Atomic Entities such as:
        - Information: profile of sensitive details of an User, Shelter or Animal. Can only be accessed by administrators or people with authority.
    - Each Database has attributes: User Profiles, Shelter Profiles, Animal Profiles, ect
6. Cloud Service: This will be the server retrieving and sending messages and notifications to Users and Shelters.
    - Composed with Atomic Entities such as:
        - Text messages: amount of words that a User sends to another.
        - Notifications: amount of words that reminds or lets that User know of something.
7. App: Satorican is an application for mobile devices and PCs.
    - Composed with Atomic Entities such as:
        - Software interface: a set of lines of code that have functions and features as an executable system.

Domain Functions

1. User
    a. getProfile(User person) = returns selected User's account information if that user exists. Otherwise, returns nothing.
    b. getProfile() = returns their own account information.
    c. setProfile() = sets their own account information.
    d. searchShelter() = returns a list of shelters near User's current location.
    e. getShelter(Shelter building) = returns selected Shelter's profile information and list of animals available for adoptions.
    f. searchAnimal() = returns a list of animals available for adoption near User's current location.
    g. getAnimal(Animal creature) = returns selected animal's profile information if it exists and is available for adoption. Otherwise, returns nothing.
    h. sendAdoptionRequest(Shelter building, Animal creature) = sends to a selected Shelter their own account information and petition to adopt a selected animal.
    i. sendAdoptionRequest(User person, Animal creature) = sends to a selected User their own account information and petition to adopt a selected animal.

j. getPet(Animal pet) = returns their pet's profile.
k. setPet(Animal pet) = sets their pet's profile.
l. sendPetTransferRequest(Animal pet, Shelter building) = sends to a selected shelter their pet's profile information and petition to give ownership. If approved by Shelter, remove from profile and send their pet to selected Shelter.
m. transferPet(Animal pet, User person) = once received and accepted an adoption request from another User, remove and send their pet to the requesting User.
n. setPetToDead(Animal pet) = remove pet from profile with a death certificate provided by a vet.

2. Shelter
    a. getShelter() = returns their profile information.
    b. getShelter(Shelter building) = returns selected Shelter's profile information.
    c. setShelter() = sets their profile information.
    d. getStaff() = returns a list of employees working in this Shelter.
    e. getUser(User person) = returns selected User's profile information.
    f. addAnimal(Animal creature) = add an animal to their list.
    g. setAnimal(Animal creature) = set an animal's profile information.
    h. acceptAdoptionRequest(User person, Animal creature) = after accepting an adoption request, remove animal from list and add it to requesting User's profile. Also, opens a chat with the requesting User.
    i. declineAdoptionRequest(User person, Animal creature) = after declining an adoption request, send a reasonable explanation to requesting User.
    j. sendAnimalTransferRequest(Shelter building, Animal creature) = sends to another Shelter an animal transfer request with the selected animal's profile.
    k. sendListOfAnimalsTransferRequest(Shelter building, ArrayList<Animal> creatures) = sends to another Shelter a list of animals transfer requests with profile information of each animal.
    l. acceptAnimalTransferRequest(Shelter building, Animal creature) = add animal to their list and remove it from requesting Shelter.
    m. declineAnimalTransferRequest(Shelter building, Animal creature) = after declining an animal transfer request, send a reasonable explanation to requesting Shelter.
    n. acceptListOfAnimalsTransferRequest(Shelter building, ArrayList<Animal> creatures) = add the list of animals in their current list and remove it from the requesting Shelter.

   o. declineListOfAnimalsTransferRequest(Shelter building, ArrayList<Animal> creatures) = after declining a list of animals transfer requests, send a reasonable explanation to requesting Shelter.
   p. setAnimalToDead(Animal creature) = remove animal from list with a death certificate provided by a vet.
3. Animal
   a. getAnimal() = returns this animal's profile information.
   b. setAnimal() = sets this animal's profile information.
   c. setAnimalToDead() = sets this animal as dead with a death certificate provided by a Vet.
4. Administrator
   a. getDatabase() = returns database.
   b. getFeedback() = returns a list of feedback from Users and Shelters.
   c. getReports() = returns a list of Reported Accounts.
   d. getUser(User person) = returns selected User's profile information.
   e. getShelter(Shelter building) = returns selected Shelter's profile information.
   f. banUser(User person) = selected User is set as banned account and can no longer function.
   g. banShelter(Shelter building) = selected Shelter is set as a banned account and can no longer function.
5. Database
   a. getData(Data info) = returns an information from the app storage.
   b. addData(Data info) = adds information to the app storage.
6. Cloud Service
   a. createChatUserWithUser(User person, User person) = create a chat between two Users.
   b. createChatUserWithShelter(User person, Shelter staff) = create a chat between a User and a staff member of selected Shelter.
   c. createChatShelterWithShelter(Shelter staff1, Shelter staff2) = create a chat between a staff from a Shelter and another staff from another Shelter.

Domain Events

1. User
    - User found a Shelter: The user can now have access to selected Shelter's profile.
      Formal Presentation: User found a Shelter
    - User found an Animal: The user has now access to information about the selected animal.
    - User received an approval on its adoption request: the user is now the owner of the animals requested. The user can now pick up its new animal.
    - User received a decline on its adoption request: the user cannot claim ownership on the requested animal with reasons found on their profile.
2. Shelter
    - Shelter received an Adoption Request: A staff gets a notification that tells them a User is interested in adopting one of their animals. Now the responding staff can view the requesting User's profile and decide whether to accept or decline their request.
    - Shelter received an Animal Transfer Request from another Shelter: A staff member gets a notification about the request. Now the responding staff can view the animal's profile and decide whenever to accept or decline the request.
    - Shelter received a List of Animals Transfer Request from another Shelter: A staff gets a notification about the request. Now the responding staff can view each animal's profile and decide whether to accept or decline the request.
    - Shelter received Animal Transfer Request from User: A staff member gets a notification about the request. The responding staff gets to view the animal's profile and decides whether to accept or decline.

Domain Behaviors

1. User
    - User adopting an animal from a selected Shelter: Users are most likely to look for an animal to own. Therefore, they will eventually be interested in adopting an animal.
    - Users transferring their pet to a shelter: Some Users may regret or be unable to take care of their pets. Therefore, they will take their pet to a shelter.
    - User adopting an animal from another User: Some Users may regret or be unable to take care of their pets. Furthermore, they will put their pets available for adoptions.
2. Shelter
    - Shelters transferring animals to another shelter: Some Shelters may be unable to take care of a certain amount of animals. Another conflict would be that some shelters may be too full of animals. Their best option could be sending some animals to another Shelter that are able to take care of them.
3. User & Shelter
    - Users or Shelters proving their pet is dead in the app. Every animal is mortal and are most likely not going to outlast their owners. They will have to submit the death certificate to justify classifying an animal as dead in the system.

Domain Requirements

- Users must have residence with power and water, certificate of good behavior and governmentally issued identification with photo.
- Users and Shelters must upload in their profile their pet’s photos and a 20 second video every three months.
- Users and Shelters must maintain their account active as long they have a registered pet in their profile.
- Shelters must be certified with documents from the government to register.

Interface Requirements

- Database must always have the latest update from each account.
- Navigation system must display a map in User's current location and the most optimal route between User and the selected Shelter.
- Navigation system must locate and display Shelters near the User's current location.
- Satorican must have access to the internet in order to operate.
- Satorican must display a list of available adoption animal's photos and profiles that are near to User's residence. While the User must swipe left or right on each photo to decide which are the animals they are interested in adopting.
- Satorican must only save videos and photos taken in real time.
- Cloud Service must connect between Users to allow them to send messages and requests to each other.
- Cloud Service must send notifications to Users about requests, messages and reminders.
- Administrators must keep maintenance after realising Satorican.

Machine Requirements

- Satorican must be compatible and downloadable to mobile devices, PCs and laptops.
- Devices using Satorican must have cameras to capture real time photos and videos.

Inconsistencies

In the process of defining the goals needed to make this application succeed, almost inevitably, inconsistencies were found. The main focus of our application is to serve as an intermediate between the potential client who's interested in adopting an animal, and the animal shelter who will provide the client with the product they desire (in this case, the animal of his/her choosing). What we have defined as the "satisfied customer stage", is that moment where the entire process is completed and the customer is now the proud owner of the pet he just adopted. Now, the word "owner", might serve as a contradiction, since some might not have any issues in seeing the customer as the owner of the pet adopted from the shelter, but others might. Even though we define "owner" as the person now in charge of taking care of the pet, others might disagree saying animals are "free spirited" and can never be "owned".

Even though we only intend to have satisfied customers who achieve what they expected to achieve and perhaps more, we understand how “ownership” can be debated between different sectors, and have no choice, but to highlight it as an inconsistency in our application.

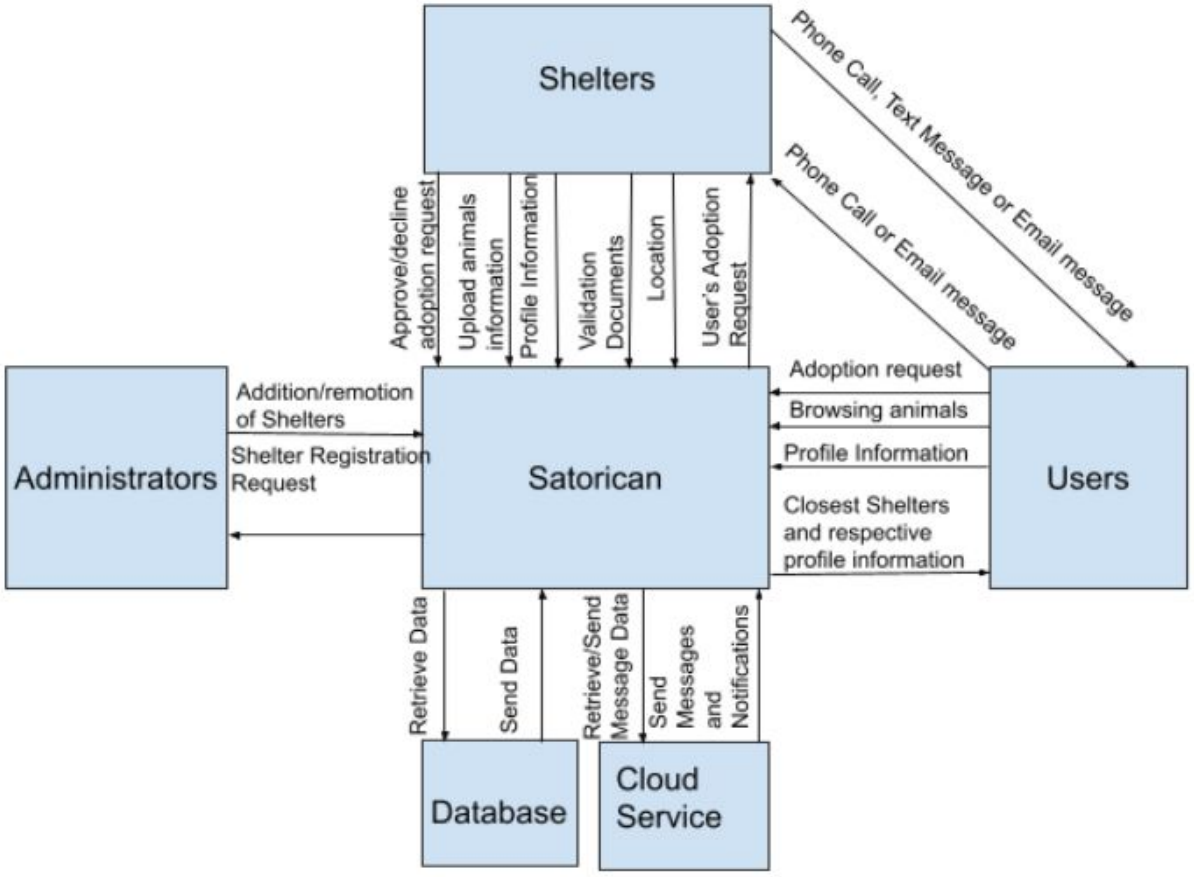
Analytical

Concept Analysis of Rough Domain Sketch

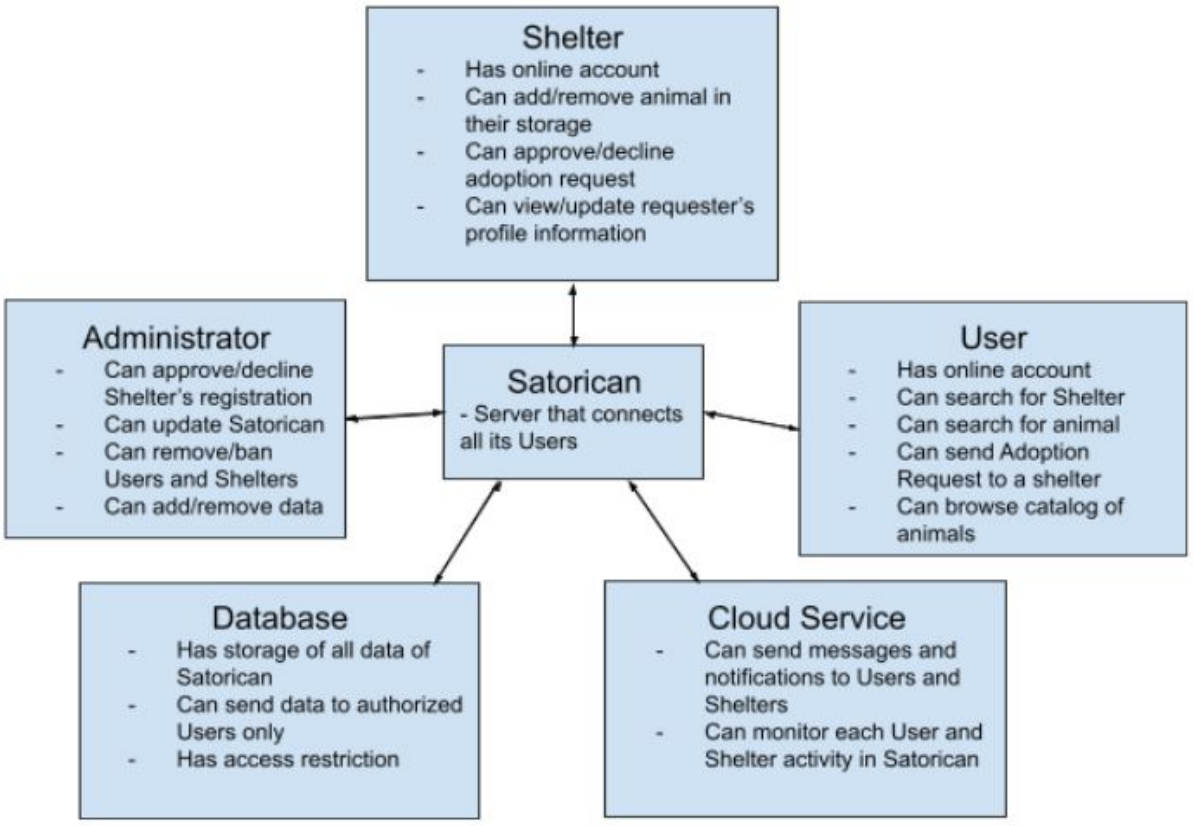
Regarding the acceptance of users into the system this should be done in a legal matter. A paper trail should be established to exempt us from legal repercussions. The basic concept for the system is to create a catalog where the user can look at all the available animals from multiple institutions from all over Puerto Rico. This activity can be divided into subcategories: browsing and searching. Those who browse simply look through all the available animals in all the institutions. While searching implies that keywords regarding specific characteristics of the animal is desired. Be it physical traits or location, the user is looking for something in specific. For the orientation and implementation of the system meetings (which contents shall be saved in a logbook) with shelters/sanctuaries that desire our services should first be established. Giving us an opportunity to further understand the domain. Once that initial meeting is set the orientation (pitch) process can occur to commence the implementation of the system.

For the retrieval and insertion of the animal’s information into the system to occur seamlessly a profile for the institutions must be created so they may upload the animals information to their page. After the user, be it through browsing or searching, finds an animal they wish to adopt a request through the animal’s page must be submitted to the institution that has it. It is then the institution's job to establish communication and evaluate the user to then decide if they are able to adopt the animal. This communication can happen through a messaging feature in our system or directly between the institution and the user. The acceptance or denial of the request must be processed through our system.

Software Architecture Design



Software Component Design



Test Plan/ Test Proposal

The main idea of our application Testing plan is to discover and resolve potential defects/bugs at each specific level in the development process. In addition, we want to prioritize different tests to make them consistent with our web application requirements and estimate a timeframe for their implementation. This is why we will focus on white box testing first to study the internal functions of the system. This will be done together with the Unit test. For this process, we intend to isolate impactful methods or classes to verify that the returns are as required or intended for this project. At the same time during this process we intend to analyze the most effective way that the method or class runs. These tests are planned for the early stages of our web application development.

Another priority is to conduct Documentation testing before and during the implementation process to track established requirements, domain, execution progress, software design, etc.Integration testing is planned for the second phase of our web application implementation. Once our diverse components are working properly, this test will help prove how they work together.At the same time, the Database test will also verify the integration of the Web application database, data consistency, whether the query is executed correctly, and whether the data is correctly retrieved and updated.

As part of the final stage of testing, Black box testing will be used to determine how the system responds to a given specific input, its response time, usability issues and reliability issues. Also, we plan to perform Retesting and Regression testing for each of the mentioned levels to ensure that every detail is implemented properly and that the entire process is consistent with our end goal.

Our future interest and ultimate goal is to enable end users to perform various testing methods (such as Acceptance testing, Usability testing and Performance testing) on our web application. In this way, we can corroborate that our application exceeds user needs and demands.

Roles:

- **Luis D. Gonzalez** - Front End / UI Design
- **Victor M. Baez** - Front End / UI Design
- **Zulmarie Jiménez** - Back End / Database integration
- **Arnaldo Villarrubia**- Back End / Database integration
- **Efraín Oliveras** - Back End
- **Angel Ortiz** - Back End