**Design Paradigm:**

Our project uses a data-structured centered design. We choose this as the main purpose of the website is to search to make searching for information in a database and adding information to that database easier for rescue organization personnel. There was really only two ways for us to go about this purpose (with the other being object oriented) and this paradigm seemed the better fit of the two as working with a variety of objects with a large number of different arrays would have quickly become cumbersome and hard to follow.

For project 3 we are only implementing the UI of the design, so we haven't yet set up the databases however when trying to design the project using the object-oriented design it quickly became too confusing. Having one centralized database with all of the information needed stored in a way that can be queried using SQL will allow the creation of the website to be far more efficient.

The entities in the database will have two main types: users and animals. Users will classify what type of user they are as well as store what animals belong to that user. The animal entity will store a large amount of information on the animal (age, health, possible adoption date, etc.) which will also include what foster (user) they are currently cared for by. Using the data-structured centered design paradigm means the website can interact directly with this data base, querying information as well as updating information as needed through the website UI.

**Software Architecture:**

The client-server architecture most closely matches the architecture of our project (as it is being created as a website). The UI of our website will allow clients to login (credentials are stored on a database server), the website will verify those credentials and will then provide access to increased functionality based on the type of user verified. Even if the user does not login to the website, they can still request to view available animals (searching with a variety of parameters possible) and the database will be contacted to collect the relevant information to then send back and display for the user.

Once a user has been verified, they will also be given a way to update information on animals assigned to them. They will be able to update information as well as add pictures. All of this information, once updated, will be uploaded to the database for other user to view (as well as be a safe location to store this data). Some of the data stored on the database will only be visibly to the rescue organization's employees while other pieces of information will be visible by anyone accessing the website (as stated in the first paragraph of this section).

We have chosen to use HTML, CSS, JavaScript, and SQL to complete this project. All of these will work together to create a website with access to a database server that will be able to accomplish the requests listed above and return the valid information to the user in a visually pleasing way.

**Design Patterns:**

We have used two main design patterns while working on our website for project 3 (one implemented in the UI) and one will be implemented in the final stages of the project. We have utilized the Decorator pattern in the implementation of the UI and will be implementing the iterator pattern in the backend access of information from the database.

We are using HTML, CSS, and JavaScript in the code of our website and all three of these uses the decorator design pattern as a fundamental aspect of its code. We have created different requirements for different wrapper objects using CSS and JavaScript. Once those parameters are set we use the HTML code to implement text and images using those wrappers to customize the website. We are using wrappers such as <button>, <p>, etc.

When we implement the backend of the website (project 4) we will be using the iterator design pattern, using SQL. The user will be able to send queries to the database using the website. Those queries will be translated into searching the database as an iterator for the information that matches the queries. The corresponding information and only information that corresponds to the query will then be relayed to the user through the website.

Of course the information relayed back to the user will also be displayed using decorator design pattern. The information will be sent through wrappers to correctly create a visually pleasing display of the information requested for ease of access as well as easy updating.