1. **Name:** Courtney Solano
2. **Title:** Animal Shelter
3. **Project Summary:** An animal shelter interface that the general public has access to view which displays the animals available for adoption. Volunteers are in charge of updating the animals that are available, and adding new animals as they show up to the shelter.

---

4. **Project Requirements:** Based on the project summary, what are the user requirements and responsibilities for your system? List the user requirements in a table format with the ID # and the user requirement. Be sure to break down the requirements into small manageable separate requirements.
   You can write the requirements in short sentences or in the Agile format.

ID	User Requirement
UR1	Customers can view animals of all type
UR2	Volunteers can log in
UR3	Volunteers can change their password
UR4	Volunteers can add animals
UR5	Customers can browse animals
UR6	Customers can search for an animal by type
UR7	Volunteers can log out
UR8	Volunteers can edit an animal's information
UR9	Volunteers can sign up using their employee id
UR10	Customers can adopt animals

---

5. **UI Mockups:** Create screen mockups for the user interface of various parts of your application.
   a. What will a user see as they work through the tasks identified in your use cases?
   b. What is the overall organization of your user interface?
   c. How will data be displayed?
   d. How will the user navigate from screen to screen?

   Note: it is okay to work on paper for this task and then scan in your work to include in your document. There are also free wire-framing tools online.

   (See file UI.pdf)

---

6. **Class Diagram:** Create a class diagram containing: what relationships the classes have, their attributes and (public) methods, what design patterns you may already know about are present in your design, etc. Be sure to show the visibility modifiers and relationships between the classes.

a. Focus your class diagram on the classes you need to create.

For classes from Java or another library: You can specify an inheritance or composition relationship <u>if it helps</u> but do <u>not</u> put all the attributes/behaviors of the framework classes into the diagram.

For example,
If I was using Java Swing I could simply specify "submitBtn : JButton" as an attribute in one of my classes without drawing the JButton class. If I want to make use of Java's Observer interface, I may draw the Observer interface in the class diagram to show a class implements it.

b. Show all classes you are coding, including attributes, methods, visibility modifiers, relationships to other classes.
Show relationships between all classes. *No floating classes.*