







User

//Anonymous at this level

Search() //Remember, browsing is just a special case of searching
donate(PetProvider)

//Instantiated at login.

UserAccount

accountID //unique code for the account (might be generated by DB)
//userName //?? Should probably use a username/password manager
firstName
lastName
City //Location information might be used in some searches.
state
zipCode
Pets[] //Pets administrated by the profile. Enables “view my pets” quickly

generateAccountID() //Might be handled by the DB
editAccountDetails() //This is the actual information for the account
editProfileAppearance() //This is just how the account displays
adopt() //The function that processes an adoption confirmation code.
//Calls a verification routine handled by the Verifier.

PetProviderAccount (type of useraccount, non-admin, such as a volunteer)

petProviderID[] //These are codes for the adoption facilities/vets, different
//from the accountID for the user.
//Why an array? They might work for more than one.

addNewPet() //Allows the creation of a pet page. Only by pet providers?
setters/getters
claimPetPage() //Sometimes the adoption center needs to forcibly claim a
//pet’s page.

PetProviderAdmAccount

addNewAdministrator(petProviderID) //Give administrator rights to a PPaccount
petProviderProfileEditor() //Edits this’s profile deets
editPetProviderProfileAppearance() //edits the actual display

VeterinarianAccount (special type of PetProvider)

animalTypes[] //?? (Some vets only treat certain types of animals)
vetID //A special ID code

//If this vet treats a pet, it should be able to look up its patients. Similar to
// “my pets” but this is shared by a vet facility.
patients[]
addMedicalRecords(petID)
editMedicalRecords(petID) //Should use a verifier (see below)

GlobalAdministrator //Has admin rights for the ENTIRE site
blockUser()
deleteAccount() //for some definitions of “delete”
deletePage()
//Needs to be able to edit any profile’s or page’s details

//This is the actual pet provider facility, not someone who uses it.
//Should be dynamically generated from DB info.

PetProvider

city //Location information used in searching.
state
zipCode
petProviderID //The unique code for this facility.

//Letting the PetProvider object keep its own pets array allows anonymous access
Pets[]

donationInfo //Don’t know what this is yet.

//Should be dynamically generated from DB info.

PetProfile

petID //A special unique code for the animal
vetID //The ID of this’s veterinarian
medicalRecords[] //??
owner //ID of the pet’s administrator
breed
color

generatePetID() //this might be a temporary thing, until the animal is chipped,
//or some animals might not be chipale and we need a way to
//keep track of them

transfer_PetPage(destinationAccountID) //A user sends this to another user
viewMedicalRecords(accountID) //Display the medical records stored in DB

```

viewFullPicture()           //must verify account has access.
petProfileEditor()         //Some script to blow up the thumbnail
getters/setters

//Even without special characteristics for the subtypes, remember that the
//concept of “is a type of” is in many OO languages; helps with search or DB
Dog
Cat
Rabbit
Bird
Other           //could be different options for each type of animal
    characteristic1 //for searching ... if you think of any, name them
    characteristic2 //or we should really make the customer name them ...
    characteristic3

```

//Instantiated when the system is loaded.

```

Verifier           //validates codes
    verifyUserAccountActivationCode(code)
    verifyPetProviderAccount(code)
    verifyAdministratorRightsCode(code)
    verifyMedicalRecordsAccess(petID, accountID)

```

Webpage //These should be HTML, so take “display()” as read.

Homepage

```

Menu/Header           //Stuff that’s initializable from any page in the domain.
    login(username, password)           //Need a secure manager ...
    createAccount()
        createPetProviderAccount()
    navigateTo...()           //Obviously just e.g. hyperlinks in the menu
    viewMyPets()             //Displays, as if search results, the profiles in this’s pets[]

```

//Dynamically generated from objects.

```

PetProviderPage           //Display mechanism
UserProfilePage           //display mechanism for the User account details

```

```

Search                   //For advanced search options
Search Results           //includes Browsing, View My Pets, etc.

```

PaymentProcessor //lives in a frame on a pet provider profile, but it's its own thing
processPayment() //This should just load the payment processor plugin
storePaymentStats() //Keep some stats like the amount donated?
processAcknowledgment() //Depends on the implementation inherent in the plugin

UserNameAndPasswordManager

verifyLogin() //Whatever this entails.

DatabaseManager

//Fill in as we get more familiar with the database

method1()

method2()