**Executive Summary**

*1.* *Show the owner of the pet (possibly none or 1)*

By clicking a specific “Owner” button, the current owner of the pet will be shown.

* updated **src/js/app.js**: modified markAdopted(), handleAdopt(), initContract()
* updated **contracts/Pets.sol**for new data structure to support new functions: added create(), idExists(), idFind(); modified adopt(), getPets()

*2. Up/down vote a pet*

By clicking the “Up Vote” or the “Down Vote” button, the user can up/down vote a pet, the current vote number will be reflected immediately.

* updated **src/js/app.js**: added markVotes(), handleUpVote(), handleDownVote(); modified init(), bindEvents()
* updated **contracts/Pets.sol**: added upVote(), downVote()

*3. Return a pet*

By clicking the “Return” button, the user can return an adopted pet.

* updated **contracts/Pets.sol** to support pet return: added returnPet()
* updated **src/js/app.js**: modified init(), bindEvents(), markAdopted() to support the behaviour of "Return" button; added handleReturnPet() to return the pet for a fee
* updated **index.html** to support the "Return" button

*4. Adoption histories*

By clicking the “Adoption History” button, the user can view the adoption history.

* updated **contracts/Pets.sol**: added adopterHistory data and collect it in the adopt() function.
* updated **src/js/app.js**: modified markAdopted() to support the behaviour of " Adoption History" button; added handleReturnPet() to return the pet for a fee
* updated **index.html** to support the "Adoption History" button.

*5. Track the number of adopted pets and served customers*

The number of served customers and adopted pets is placed under the title of the Pet Shop, allowing the user to know the popularity of the shop.

* updated **contracts/Pets.sol**: added addCust() to track customers, trackPet() and trackCust() to get how many customers have been served and how many pets adopted
* updated **src/js/app.js**: add markPets(), markCusts()to support the behavior; modified markVotes() handleAdopt() to trigger the behaviour
* updated **index.html** to support information display

*6. Use “@truffle/contract” instead of “truffle contract”*.

* updated **index.html** and **bs-config.json**
* deleted **src/js/truffle-contract.js** and **src/js/web3.min.js**

*7. Use npm “bootstrap” resources instead of local “bootstrap” resources*

* updated **index.html** and **bs-config.json**
* deleted **src/js/bootstrap.min.js**, **src/css/bootstrap.min.css**, and **src/css/bootstrap.min.css.map**