

Dog Profile Test Cases

Test case 1 (for Dog Expanded Info): Test if Dog Profile Scene is set up correctly after logging in.

Steps:

1. When the next scene loads, the GUI is expected to resemble the screenshot below:
- A central dog picture (picture may vary) with a "like" button to its left and a "pass" button to its right.

- The first row beneath the picture should display the dog's name, age, and sex accurately.

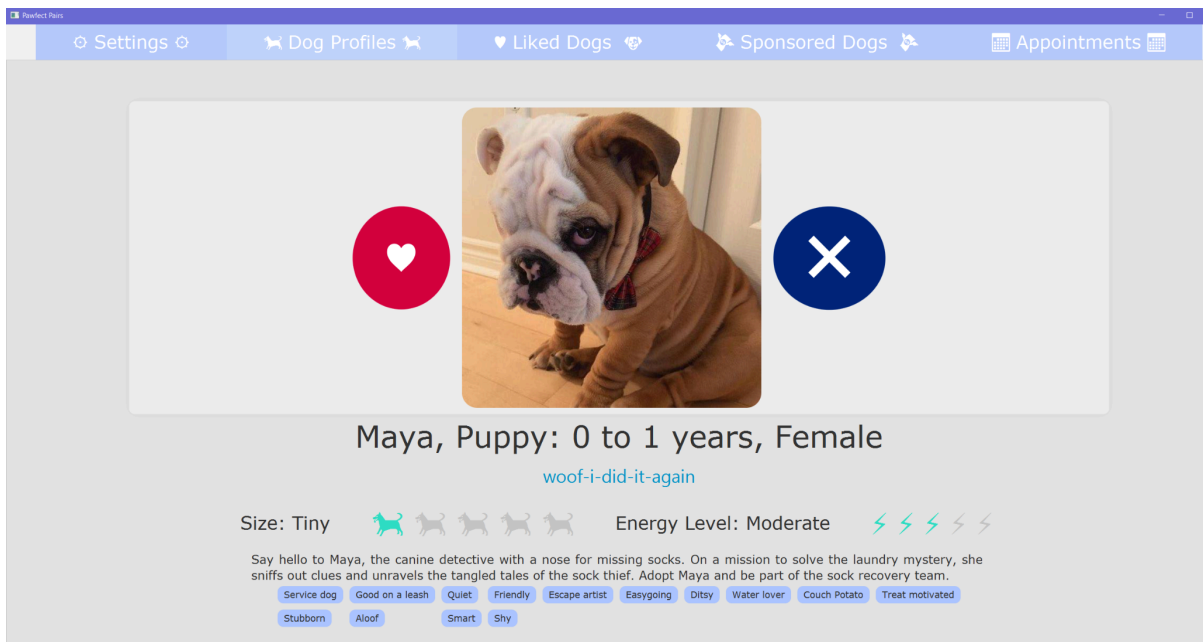
- Following that, the poster's (dog owner's) name should be shown in the next row.

- Dog size and energy level should be represented by icons out of a scale of 5.

- The dog's biography will then appear below this.

- Lastly, individual text boxes will display tags describing the dog at the very bottom.

(the example shown in the screenshot is Maya (dogID:76))



2. The information displayed on this dog profile is expected to match the information of this dog in database (the example shown in the screenshot is Maya (dogID:76)):

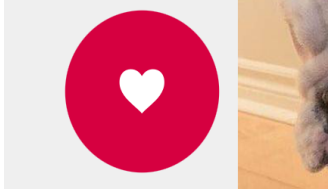
	<b>dogname</b> character varying (50)	<b>adopted</b> boolean	<b>biography</b> character varying (1000)			
74	Chance	false	Chance is a retired rockstar, known for howling chart-topping tunes in the shower. With dreams c			
75	Brandy	false	Say hello to Brandy, the neighborhood's canine DJ. With a playlist featuring the best squeaky toy:			
76	Maya	false	Say hello to Maya, the canine detective with a nose for missing socks. On a mission to solve the			

<b>imagepath</b> character varying (1000)	<b>ageid</b> integer	<b>energylevelid</b> integer	<b>sizeid</b> integer	<b>sexid</b> integer	<b>posterid</b> integer	<b>dogid</b> [PK] integer
file:src/guilayout/images/dog124.jpg	4	2	0	1	8	74
file:src/guilayout/images/dog17.jpg	0	2	0	1	19	75
file:src/guilayout/images/dog30.jpg	0	2	0	1	17	76

**Test case 2 (for Liking Functionality): Upon clicking the “Like” button, the dog profile scene should automatically show the user the next dog.**

Steps:

1. In the Dog Profile scene, click the “Like” button (the red circle with a white heart in the middle on the left side of the dog picture).



2. The current “liked” dog is expected to disappear and a new dog should show up on the Dog Profile scene.
  - The top dog (‘peek()’) from allDogs is added to a list of “liked” dogs associated with a user.

```
Button likeButton = Components.button("♥");
likeButton.getStyleClass().add("like-button");
likeButton.setOnAction(e -> {
    allDogs.peek().setAdopted(true);
    user.addLikedDogs(allDogs.peek());
    if (allDogs.size() == 1) {
        changeProfile();
        outOfDogs.start(primaryStage);
    } else {
        changeProfile();
        displayCurrentPetProfile();
    }
});
```

**Test case 3 (for Pass Functionality): Upon clicking the “pass” button, the dog profile scene should automatically show the user the next dog.**

Steps:

1. In the Dog Profile scene, click the “Pass” button (the dark blue circle with a cross in the middle on the right side of the dog picture).



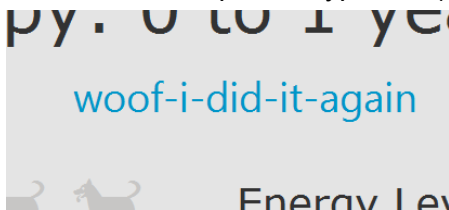
2. The current “passed” dog is expected to disappear and a new dog should show up on the Dog Profile scene.
  - The top dog (‘peek()’) from the allDogs collection is added to a list of “passed” dogs associated with the user.

```
Button passButton = Components.button("X");
passButton.getStyleClass().add("pass-button");
passButton.setOnAction(event -> {
    user.addPassedDogs(allDogs.peek());
    if (allDogs.size() == 1) {
        changeProfile();
        outOfDogs.start(primaryStage);
    } else {
        changeProfile();
        displayCurrentPetProfile();
    }
});
```

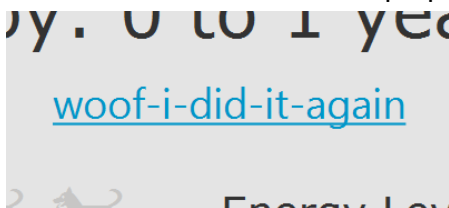
**Test case 4 (for Dog Expanded Info): Upon clicking the poster hyperlink, the user should be automatically led to the poster profile.**

Steps:

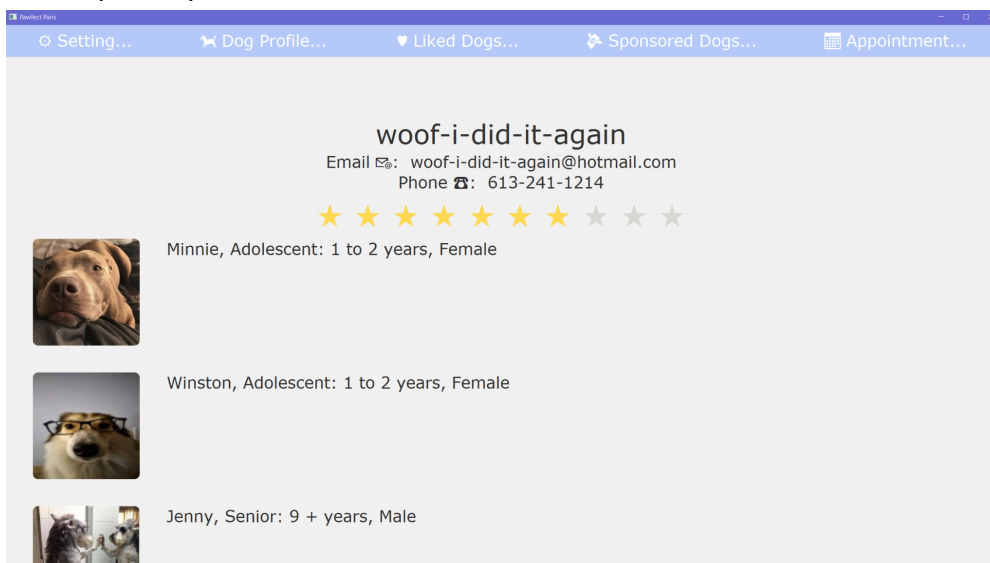
1. Click the poster hyperlink (blue text) that is underneath the dog picture.



2. When the cursor hovers over the hyperlink, it is expected to be underlined to show that it is clickable and properly linked.



3. Click the poster hyperlink, then the user is expected to be led to the corresponding poster profile.



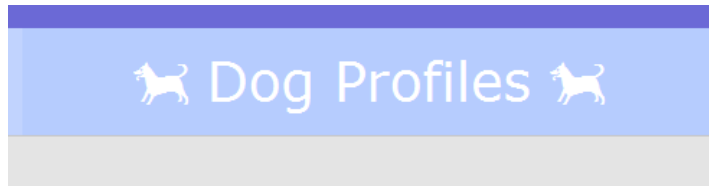
4. The displayed poster information is expected to match the information stored in the database to ensure the correct poster profile is displayed for the user.

17	17	woof-i-did-it-again	7	613-241-1214	woof-i-did-it-again@hotmail.com	0
----	----	---------------------	---	--------------	---------------------------------	---

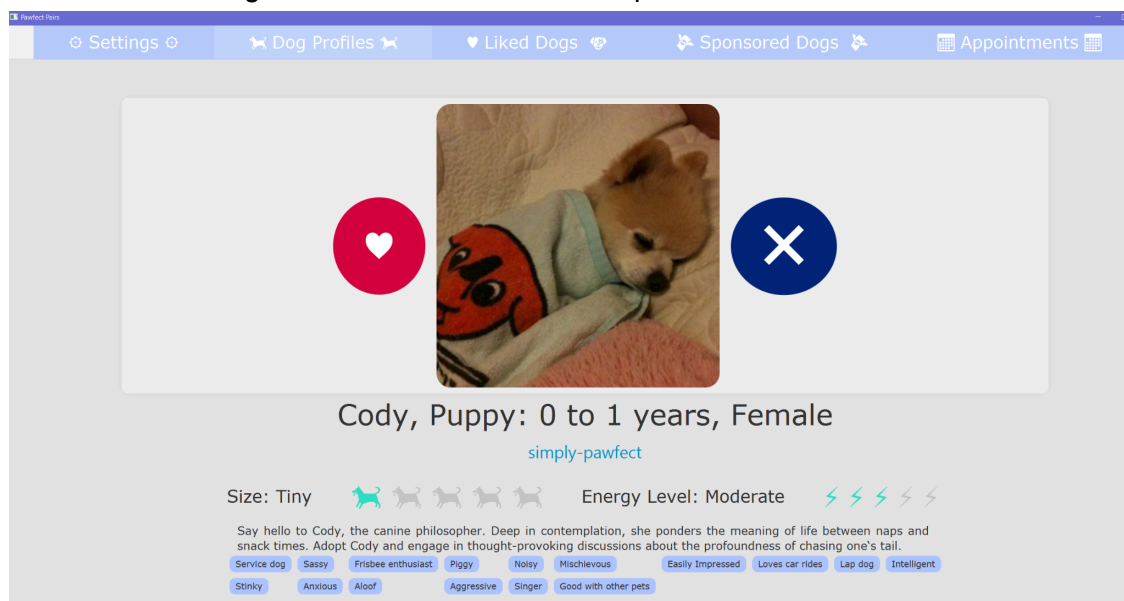
### Test case 5 (for Dog Profile): Test if “Dog Profile” button leads the user to dog profile.

Steps:

1. When the user is in another scene of the program (for example, Settings), click the “Dog Profile” button (the second one on the left).



2. After clicking the “Dog Profile” button, dog-related data is expected to be updated and the Dog Profile scene should show up.



```
dogProfileButton.setOnMouseClicked(event -> {
    appData.updateDogScores();

    dogProfile.start(stage);
});
```

### Test case 6 (for Liking and Pass Functionality): After logging in again, “liked” dog and “disliked” dog should not appear in dog profiles again.

Steps:

1. Follow step 1-2 in initialization, then type in the same Username and Password that were used last time. Then click “Login”.
2. After the Dog Profile Scene loads, the dog that has been liked or disliked by the user is expected to disappear in the dog profile. The dogs that are being displayed to the user are dogs that have not been “liked” or “passed”.

```
Button likeButton = Components.button("♥");
likeButton.getStyleClass().add("like-button");
likeButton.setOnAction(e -> {
    allDogs.peek().setAdopted(true);
    user.addLikedDogs(allDogs.peek());
    if (allDogs.size() == 1) {
        changeProfile();
        outOfDogs.start(primaryStage);
    } else {
        changeProfile();
        displayCurrentPetProfile();
    }
});
```

```
Button passButton = Components.button("×");
passButton.getStyleClass().add("pass-button");
passButton.setOnAction(event -> {
    user.addPassedDogs(allDogs.peek());
    if (allDogs.size() == 1) {
        changeProfile();
        outOfDogs.start(primaryStage);
    } else {
        changeProfile();
        displayCurrentPetProfile();
    }
});
```

**Test case 7 (for Liking Functionality):** Test if after all the dogs have been “liked”, “Out of Dogs” scene is expected to automatically show up in the dog profiles.

Steps:

1. In the Dog Profile scene, click the “Like” button to “like” every dog that shows up.
2. Once all the dogs have been “liked”, “Out of Dogs” scene is expected to automatically show up in the dog profiles.
  - If there's only one dog left in the allDogs collection:
    - The changeProfile() method is called.
    - A new stage named outOfDogs is started on the primary stage.

```
Button likeButton = Components.button("♥");
likeButton.getStyleClass().add("like-button");
likeButton.setOnAction(e -> {
    allDogs.peek().setAdopted(true);
    user.addLikedDogs(allDogs.peek());
    if (allDogs.size() == 1) {
        changeProfile();
        outOfDogs.start(primaryStage);
    } else {
        changeProfile();
        displayCurrentPetProfile();
    }
});
```

**Test case 8 (for Passing Functionality):** Test if after all the dogs have been “passed”, “Out of Dogs” scene is expected to automatically show up in the dog profiles.

Steps:

1. In the Dog Profile scene, click the “Pass” button to “pass” every dog that shows up.

2. Once all the dogs have been “passed”, “Out of Dogs” scene is expected to automatically show up in the dog profiles.
  - If there's only one dog left in the allDogs collection:
    - The changeProfile() method is called.
    - A new stage named outOfDogs is started on the primary stage.

```
Button passButton = Components.button("X");
passButton.getStyleClass().add("pass-button");
passButton.setOnAction(event -> {
    user.addPassedDogs(allDogs.peek());
    if (allDogs.size() == 1) {
        changeProfile();
        outOfDogs.start(primaryStage);
    } else {
        changeProfile();
        displayCurrentPetProfile();
    }
});
```

**Test case 9 ( for Liking and Passing Functionality):** Test if after all the dogs have been “liked” or “passed”, “Out of Dogs” scene is expected to automatically show up in the dog profiles.

Steps:

1. In the Dog Profile scene, click the “Like” or “Pass” button to “like” or “pass” every dog that shows up base on your preference.
2. Once all the dogs have been “liked” or “passed”, “Out of Dogs” scene is expected to automatically show up in the dog profiles.
  - If there's only one dog left in the allDogs collection:
    - The changeProfile() method is called.
    - A new stage named outOfDogs is started on the primary stage.

**Test case 10 (for Dog Profile):** After logging in again, “Out of Dogs” scene should still be in the dog profiles after all the dogs have been “liked” or “passed”.

Steps:

1. Follow step 1-2 in initialization, then type in the same Username and Password that were used last time. Then click “Login”.
2. After the Dog Profile Scene loads, the “Out of Dogs” scene should automatically show up in the dog profiles.
  - If there are no dogs left in the selection:
    - outOfDogs.start(primaryStage) is executed to start a new stage (window) named outOfDogs.

```
if (allDogs.size() == 0) {
    outOfDogs.start(stage);
} else {
    primaryStage.show();
}
```

Yuqian Liu

**Result:**

All test cases passed. No bugs or issues are found during the end-2-end testing. All functions including buttons and hyperlinks are working correctly. All information on the Dog Profile is displayed correctly to the user.