Final Presentation:

Animal Farm

Christopher Gonzalez, Elizabeth-Agnes Gaw,
Kathryn Saldivar, Dulce Meza-Flores

Purpose & focus this semester- Elizabeth

- Create an application to record agricultural activity
- Create a login & register system
- Tailor the website to Doc’s requirements

Tools – Kathryn

- 000webhost
- Google Meetings
- Materialize Framework
- Notepad++
- Google API Pie Chart

Github Projects & Issue Tracker

- GitHub

Functional Requirements-Kathryn

Our most valuable FRQ:

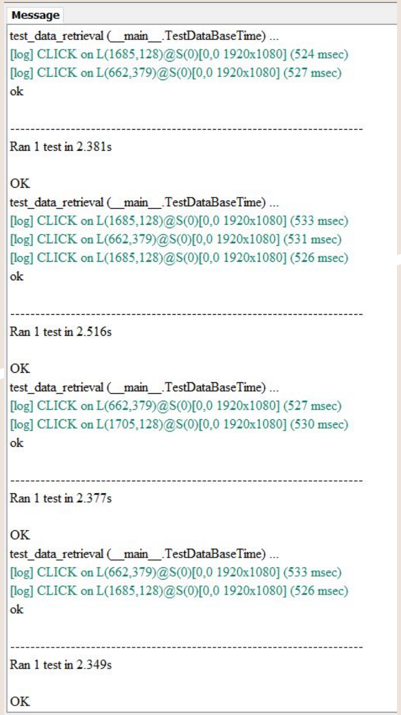
- **FR 3, 4, 5**: user shall view, add, search, edit data
- **FR 6**: user shall search and select data to edit
- **FR 8**: user shall view heat checks
- ***FR 11**: user shall be able to register with the site
- ***FR 12**: user shall be able to log in and log out of the website
- ***FR 13**: Each user shall have their own workspace
- ***FR 14**: user shall be able to view a data analysis of the breeds for all data entries

Animal Farm

Non-Functional Requirements (Kathryn)

- **NFR 1**: Fast Response time - no more than 3 seconds to load

- **NFR 2**: Easy navigation



```
test_data_retrieval (__main__.TestDataBaseTime) ...
[log] CLICK on L(1685,128)@S(0)[0,0 1920x1080] (524 msec)
[log] CLICK on L(662,379)@S(0)[0,0 1920x1080] (527 msec)
ok

----------------------------------------------------------------------
Ran 1 test in 2.381s

OK
test_data_retrieval (__main__.TestDataBaseTime) ...
[log] CLICK on L(1685,128)@S(0)[0,0 1920x1080] (533 msec)
[log] CLICK on L(662,379)@S(0)[0,0 1920x1080] (531 msec)
[log] CLICK on L(1685,128)@S(0)[0,0 1920x1080] (526 msec)
ok

----------------------------------------------------------------------
Ran 1 test in 2.516s

OK
test_data_retrieval (__main__.TestDataBaseTime) ...
[log] CLICK on L(662,379)@S(0)[0,0 1920x1080] (527 msec)
[log] CLICK on L(1705,128)@S(0)[0,0 1920x1080] (530 msec)
ok

----------------------------------------------------------------------
Ran 1 test in 2.377s

OK
test_data_retrieval (__main__.TestDataBaseTime) ...
[log] CLICK on L(662,379)@S(0)[0,0 1920x1080] (533 msec)
[log] CLICK on L(1685,128)@S(0)[0,0 1920x1080] (526 msec)
ok

----------------------------------------------------------------------
Ran 1 test in 2.349s

OK
```

average time of 2.453 seconds

Design Cont..-Dulce

Issues faced this semester:

- Needed to give data more meaning
- Restrict access to important pages
- Each user should only be able to manipulate data that *they* added

How we solved them:

- Data analysis
- login/registration system
- Each user will have their own workspace based on login

Design-Dulce

- Centralized database
- Generic application architecture
  - Adjust to our needs→ software reuse

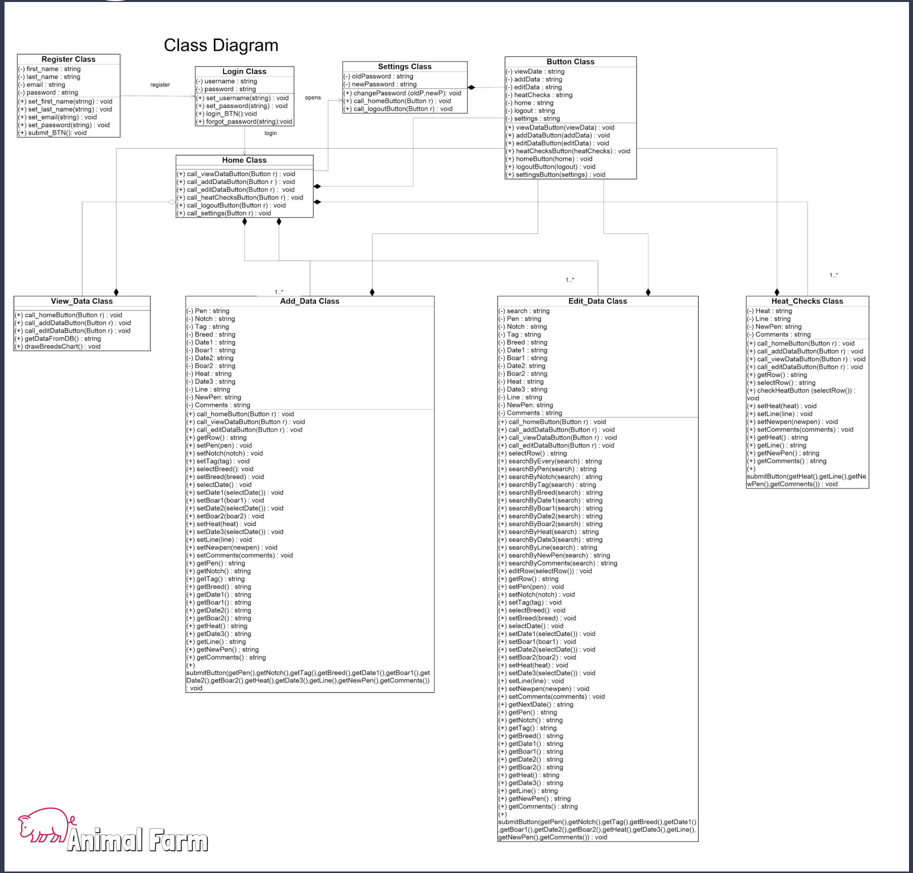
★ Layered architecture

★ Bottom Up Design and Sandwich

➢ Low coupling
  - Control coupling
  - Stamp coupling

➢ High cohesion
  - Functional cohesion

Class Diagram – Christopher



Register Class
(-) first_name : string
(-) last_name : string
(-) email : string
(-) password : string
(+) set_first_name(string) : void
(+) set_last_name(string) : void
(+) set_email(string) : void
(+) set_password(string) : void
(+) submit_BTN(): void

Login Class
(-) username : string
(-) password : string
(+) set_username(string) : void
(+) set_password(string) : void
(+) login_BTN():void
(+) forgot_password(string):void

Settings Class
(-) oldPassword : string
(-) newPassword : string
(+) changePassword (oldP,newP): void
(+) call_homeButton(Button r) : void
(+) call_logoutButton(Button r) : void

Button Class
(-) viewDate : string
(-) addData : string
(-) editData ; string
(-) heatChecks : string
(-) home : string
(-) logout : string
(-) settings : string
(+) viewDataButton(viewData) : void
(+) addDataButton(addData) : void
(+) editDataButton(editData) : void
(+) heatChecksButton(heatChecks) : void
(+) homeButton(home) : void
(+) logoutButton(logout) : void
(+) settingsButton(settings) : void

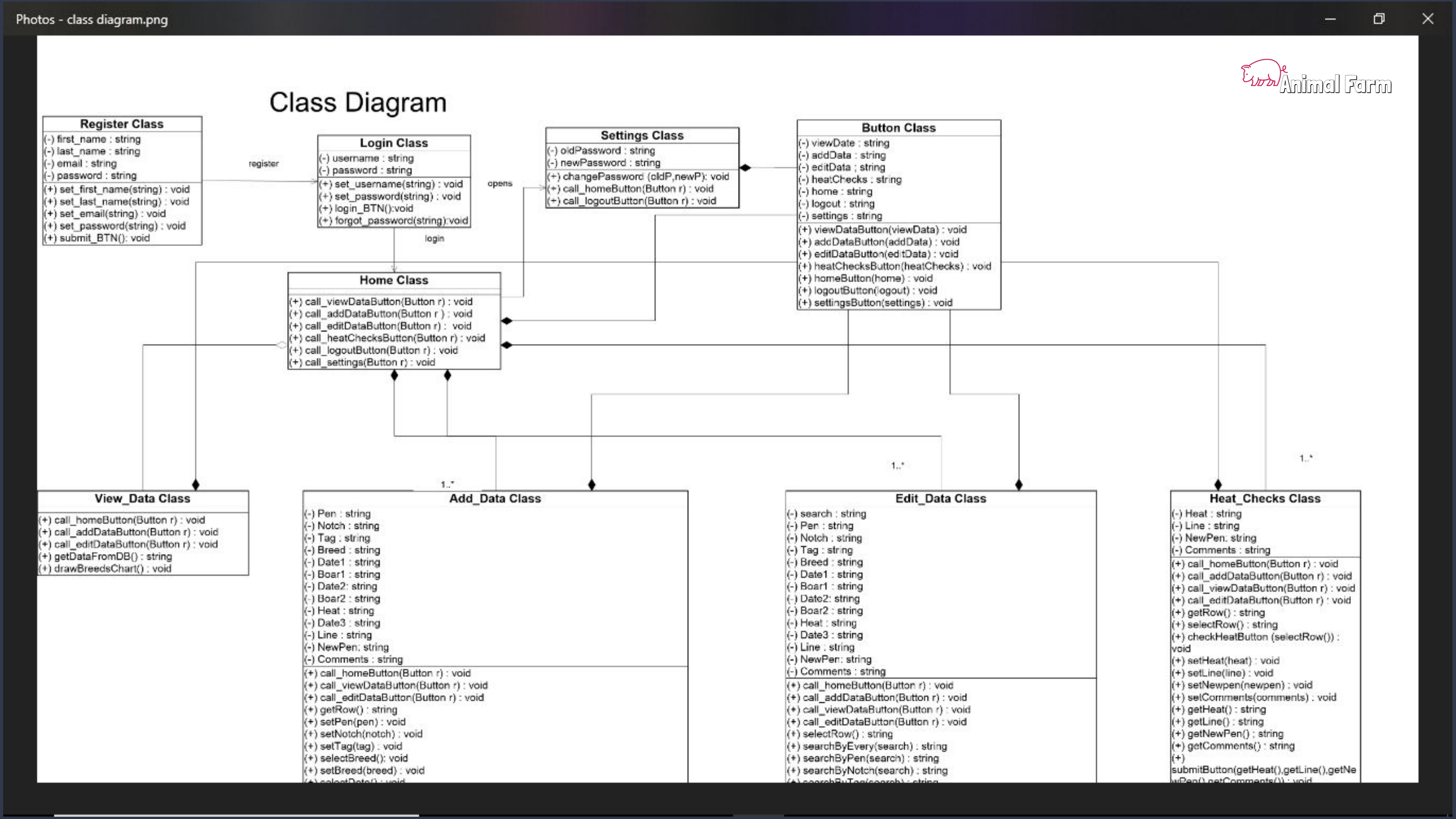
Home Class
(+) call_viewDataButton(Button r) : void
(+) call_addDataButton(Button r) : void
(+) call_editDataButton(Button r) : void
(+) call_heatChecksButton(Button r) : void
(+) call_logoutButton(Button r) : void
(+) call_settings(Button r) : void

View_Data Class
(+) call_homeButton(Button r) : void
(+) call_addDataButton(Button r) : void
(+) call_editDataButton(Button r) : void
(+) getDataFromDB() : string
(+) drawBreedsChart() : void

Add_Data Class
(-) Pen : string
(-) Notch : string
(-) Tag : string
(-) Breed : string
(-) Date1 : string
(-) Boar1 : string
(-) Date2: string
(-) Boar2 : string
(-) Heat : string
(-) Date3 : string
(-) Line : string
(-) NewPen: string
(-) Comments : string
(+) call_homeButton(Button r) : void
(+) call_viewDataButton(Button r) : void
(+) call_editDataButton(Button r) : void
(+) getRow() : string
(+) setPen(pen) : void
(+) setNotch(notch) : void
(+) setTag(tag) : void
(+) selectBreed(): void
(+) setBreed(breed) : void
(+) selectDate() : void
(+) setDate1(selectDate()) : void
(+) setBoar1(boar1) : void
(+) setDate2(selectDate()) : void
(+) setBoar2(boar2) : void
(+) setHeat(heat) : void
(+) setDate3(selectDate()) : void
(+) setLine(line) : void
(+) setNewpen(newpen) : void
(+) setComments(comments) : void
(+) getPen() : string
(+) getNotch() : string
(+) getTag() : string
(+) getBreed() : string
(+) getDate1() : string
(+) getBoar1() : string
(+) getDate2() : string
(+) getBoar2() : string
(+) getHeat() : string
(+) getDate3() : string
(+) getLine() : string
(+) getNewPen() ; string
(+) getComments() : string
(+) submitButton(getPen(),getNotch(),getTag(),getBreed(),getDate1(),getBoar1(),getDate2(),getBoar2(),getHeat(),getDate3(),getLine(),getNewPen(),getComments()) : void

Edit_Data Class
(-) search : string
(-) Pen : string
(-) Notch : string
(-) Tag : string
(-) Breed : string
(-) Date1 : string
(-) Boar1 : string
(-) Date2: string
(-) Boar2 : string
(-) Heat : string
(-) Date3 : string
(-) Line : string
(-) NewPen: string
(-) Comments : string
(+) call_homeButton(Button r) : void
(+) call_addDataButton(Button r) : void
(+) call_viewDataButton(Button r) : void
(+) call_editDataButton(Button r) : void
(+) selectRow() : string
(+) searchByEvery(search) : string
(+) searchByPen(search) : string
(+) searchByNotch(search) : string
(+) searchByTag(search) : string
(+) searchByBreed(search) : string
(+) searchByDate1(search) : string
(+) searchByBoar1(search) : string
(+) searchByDate2(search) : string
(+) searchByBoar2(search) : string
(+) searchByHeat(search) : string
(+) searchByDate3(search) : string
(+) searchByLine(search) : string
(+) searchByNewPen(search) : string
(+) searchByComments(search) : string
(+) editRow(selectRow()) : void
(+) getRow() : string
(+) setPen(pen) : void
(+) setNotch(notch) : void
(+) setTag(tag) : void
(+) selectBreed(): void
(+) setBreed(breed) : void
(+) selectDate() : void
(+) setDate1(selectDate()) : void
(+) setBoar1(boar1) : void
(+) setDate2(selectDate()) : void
(+) setBoar2(boar2) : void
(+) setHeat(heat) : void
(+) setDate3(selectDate()) : void
(+) setLine(line) : void
(+) setNewpen(newpen) : void
(+) setComments(comments) : void
(+) getNextDate() : string
(+) getPen() : string
(+) getNotch() : string
(+) getTag() : string
(+) getBreed() : string
(+) getDate1() : string
(+) getBoar1() : string
(+) getDate2() : string
(+) getBoar2() : string
(+) getHeat() : string
(+) getDate3() : string
(+) getLine() : string
(+) getNewPen() ; string
(+) getComments() : string
(+) submitButton(getPen(),getNotch(),getTag(),getBreed(),getDate1(),getBoar1(),getDate2(),getBoar2(),getHeat(),getDate3(),getLine(),getNewPen(),getComments()) : void

Heat_Checks Class
(-) Heat : string
(-) Line : string
(-) NewPen: string
(-) Comments : string
(+) call_homeButton(Button r) : void
(+) call_addDataButton(Button r) : void
(+) call_viewDataButton(Button r) : void
(+) call_editDataButton(Button r) : void
(+) getRow() : string
(+) selectRow() : string
(+) checkHeatButton (selectRow()) : void
(+) setHeat(heat) : void
(+) setLine(line) : void
(+) setNewpen(newpen) : void
(+) setComments(comments) : void
(+) getHeat() : string
(+) getLine() : string
(+) getNewPen() ; string
(+) getComments() : string
(+) submitButton(getHeat(),getLine(),getNewPen(),getComments()) : void



Class Diagram

Register Class

(-) first_name : string
(-) last_name : string
(-) email : string
(-) password : string

(+) set_first_name(string) : void
(+) set_last_name(string) : void
(+) set_email(string) : void
(+) set_password(string) : void
(+) submit_BTN(): void

Login Class

(-) username : string
(-) password : string

(+) set_username(string) : void
(+) set_password(string) : void
(+) login_BTN():void
(+) forgot_password(string):void

Settings Class

(-) oldPassword : string
(-) newPassword : string

(+) changePassword (oldP,newP): void
(+) call_homeButton(Button r) : void
(+) call_logoutButton(Button r) : void

Button Class

(-) viewDate : string
(-) addData : string
(-) editData ; string
(-) heatChecks : string
(-) home : string
(-) logout : string
(-) settings : string

(+) viewDataButton(viewData) : void
(+) addDataButton(addData) : void
(+) editDataButton(editData) : void
(+) heatChecksButton(heatChecks) : void
(+) homeButton(home) : void
(+) logoutButton(logout) : void
(+) settingsButton(settings) : void

Home Class

(+) call_viewDataButton(Button r) : void
(+) call_addDataButton(Button r) : void
(+) call_editDataButton(Button r) : void
(+) call_heatChecksButton(Button r) : void
(+) call_logoutButton(Button r) : void
(+) call_settings(Button r) : void

View_Data Class

(+) call_homeButton(Button r) : void
(+) call_addDataButton(Button r) : void
(+) call_editDataButton(Button r) : void
(+) getDataFromDB() : string
(+) drawBreedsChart() : void

Add_Data Class

(-) Pen : string
(-) Notch : string
(-) Tag : string
(-) Breed : string
(-) Date1 : string
(-) Boar1 : string
(-) Date2: string
(-) Boar2 : string
(-) Heat : string
(-) Date3 : string
(-) Line : string
(-) NewPen: string
(-) Comments : string

(+) call_homeButton(Button r) : void
(+) call_viewDataButton(Button r) : void
(+) call_editDataButton(Button r) : void
(+) getRow() : string
(+) setPen(pen) : void
(+) setNotch(notch) : void
(+) setTag(tag) : void
(+) selectBreed(): void
(+) setBreed(breed) : void
(+) selectDate() : void

Edit_Data Class

(-) search : string
(-) Pen : string
(-) Notch : string
(-) Tag : string
(-) Breed : string
(-) Date1 : string
(-) Boar1 : string
(-) Date2: string
(-) Boar2 : string
(-) Heat : string
(-) Date3 : string
(-) Line : string
(-) NewPen: string
(-) Comments : string

(+) call_homeButton(Button r) : void
(+) call_addDataButton(Button r) : void
(+) call_viewDataButton(Button r) : void
(+) call_editDataButton(Button r) : void
(+) selectRow() : string
(+) searchByEvery(search) : string
(+) searchByPen(search) : string
(+) searchByNotch(search) : string
(+) searchByTag(search) : string

Heat_Checks Class

(-) Heat : string
(-) Line : string
(-) NewPen: string
(-) Comments : string

(+) call_homeButton(Button r) : void
(+) call_addDataButton(Button r) : void
(+) call_viewDataButton(Button r) : void
(+) call_editDataButton(Button r) : void
(+) getRow() : string
(+) selectRow() : string
(+) checkHeatButton (selectRow()) : void
(+) setHeat(heat) : void
(+) setLine(line) : void
(+) setNewpen(newpen) : void
(+) setComments(comments) : void
(+) getHeat() : string
(+) getLine() : string
(+) getNewPen() ; string
(+) getComments() : string
(+) submitButton(getHeat(),getLine(),getNewPen(),getComments()) : void

Implementation - Elizabeth

1. Cool algorithms

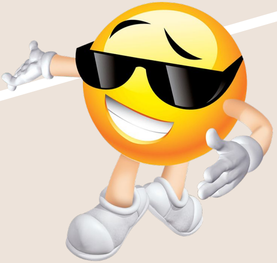
    a. Data Analysis → breed break down

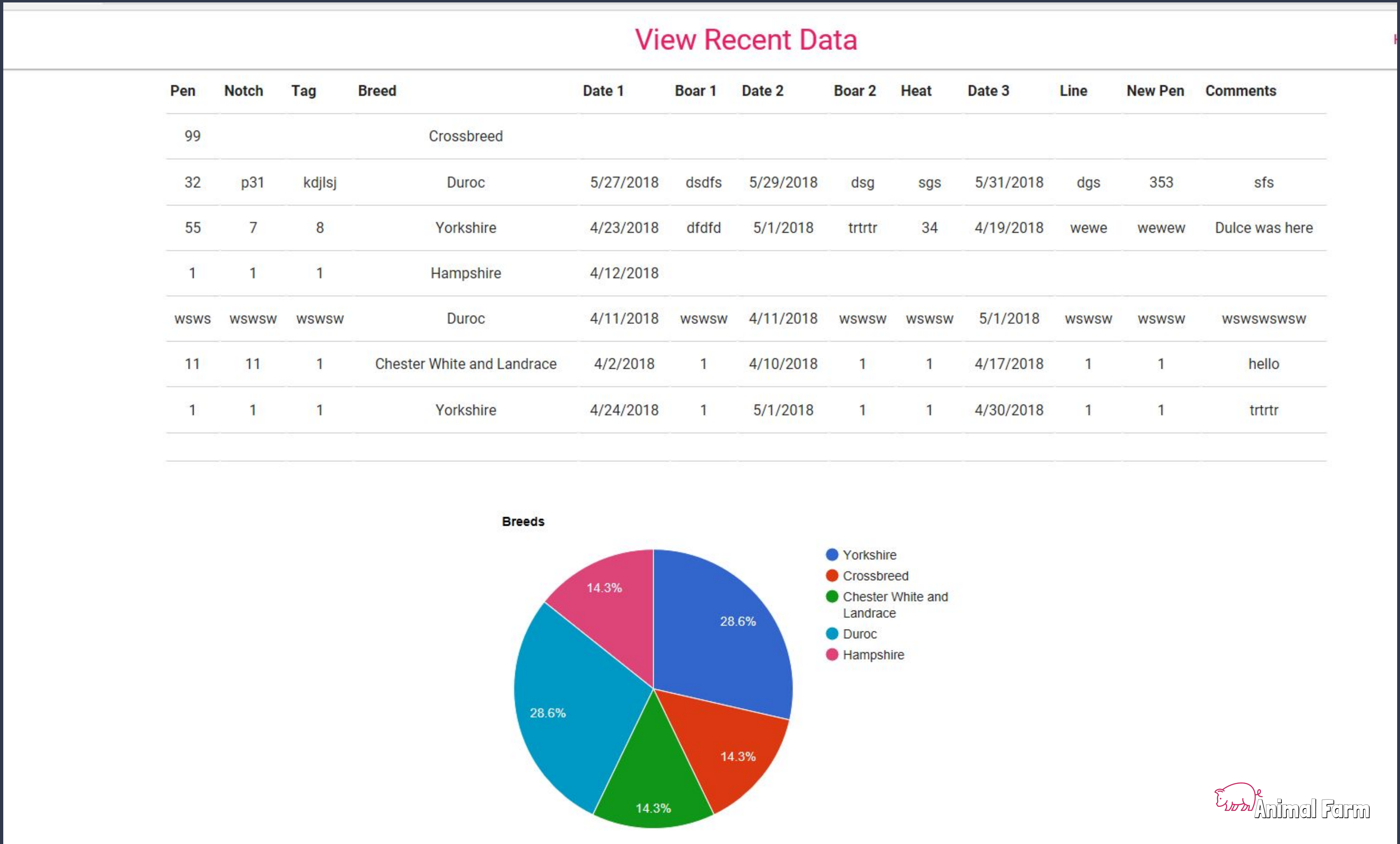
        i. Google API

        ii. Unique color for each breed

    b. Login/registration system → individual workspace

        i. Individual database





View Recent Data

Pen	Notch	Tag	Breed	Date 1	Boar 1	Date 2	Boar 2	Heat	Date 3	Line	New Pen	Comments
99			Crossbreed									
32	p31	kdjlsj	Duroc	5/27/2018	dsdfs	5/29/2018	dsg	sgs	5/31/2018	dgs	353	sfs
55	7	8	Yorkshire	4/23/2018	dfdfd	5/1/2018	trtrtr	34	4/19/2018	wewe	wewew	Dulce was here
1	1	1	Hampshire	4/12/2018								
wsws	wswsw	wswsw	Duroc	4/11/2018	wswsw	4/11/2018	wswsw	wswsw	5/1/2018	wswsw	wswsw	wswswswsw
11	11	1	Chester White and Landrace	4/2/2018	1	4/10/2018	1	1	4/17/2018	1	1	hello
1	1	1	Yorkshire	4/24/2018	1	5/1/2018	1	1	4/30/2018	1	1	trtrtr

Animal Farm

Implementation Cont.. - Elizabeth

1. Issues
   a. Difficulties with FireFox's and Google's algorithm
      i. Website marked as phishing site

   b. Login/registration
      i. Tailored the code to our application

   c. DbUnit and testing (generally)

Testing

Testing: Tools - Kathryn

- Sikuli
- Alpha Tests
- Mocha and Chai
- PHPunit and DBunit

Sikuli Script

PHPUnit

mocha

chai

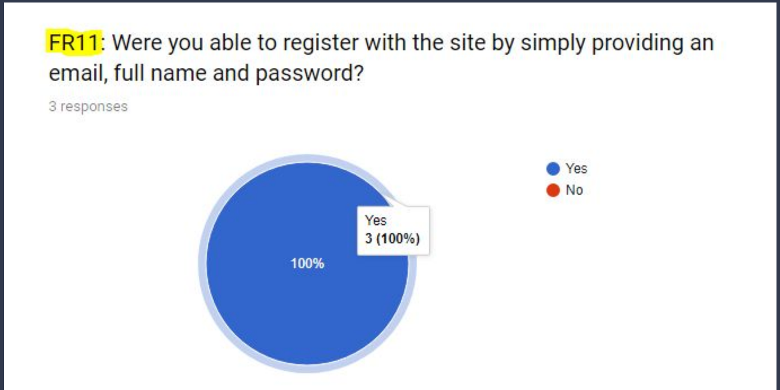
Black-Box Testing

Testing: Black-box Techniques (Dulce)

1. GUI Testing
   - Feedback from Swine Unit (alpha testing)
   - Sikuli

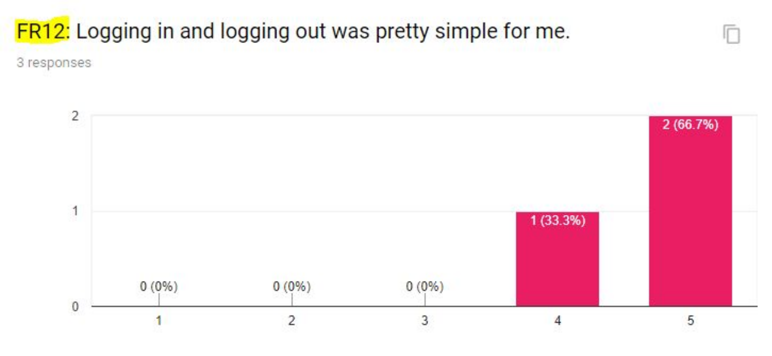
2. Cause-effect graph
   - Sikuli script using decision table test case

Alpha Testing Results–The Good 😇 (Dulce)



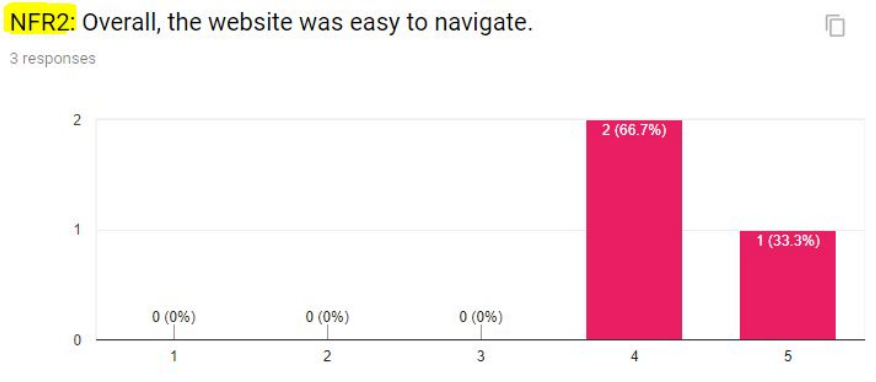
FR11: Were you able to register with the site by simply providing an email, full name and password?

3 responses



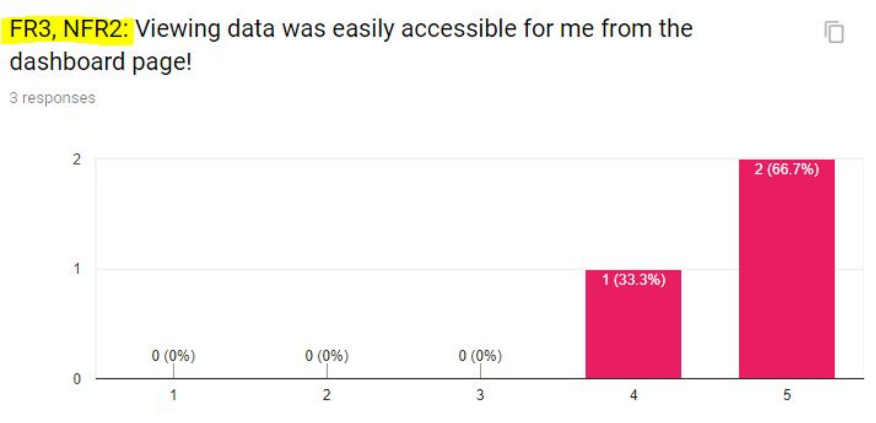
FR12: Logging in and logging out was pretty simple for me.

3 responses



NFR2: Overall, the website was easy to navigate.

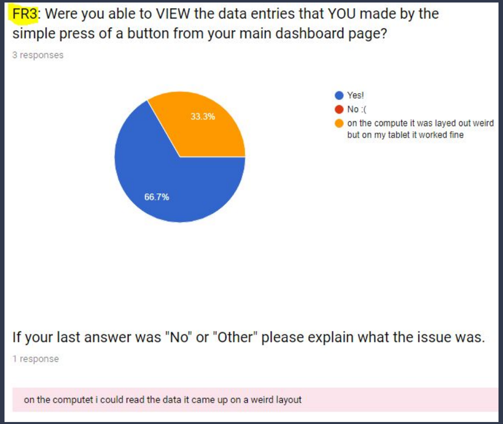
3 responses



FR3, NFR2: Viewing data was easily accessible for me from the dashboard page!

3 responses

Alpha Testing Results–The Bad 😈 (Dulce)



FR3: Were you able to VIEW the data entries that YOU made by the simple press of a button from your main dashboard page?

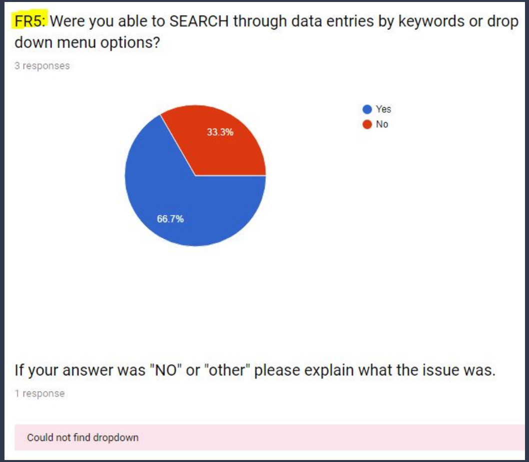
3 responses

33.3%
66.7%
● Yes!
● No :(
● on the compute it was layed out weird but on my tablet it worked fine

If your last answer was "No" or "Other" please explain what the issue was.

1 response

on the computet i could read the data it came up on a weird layout



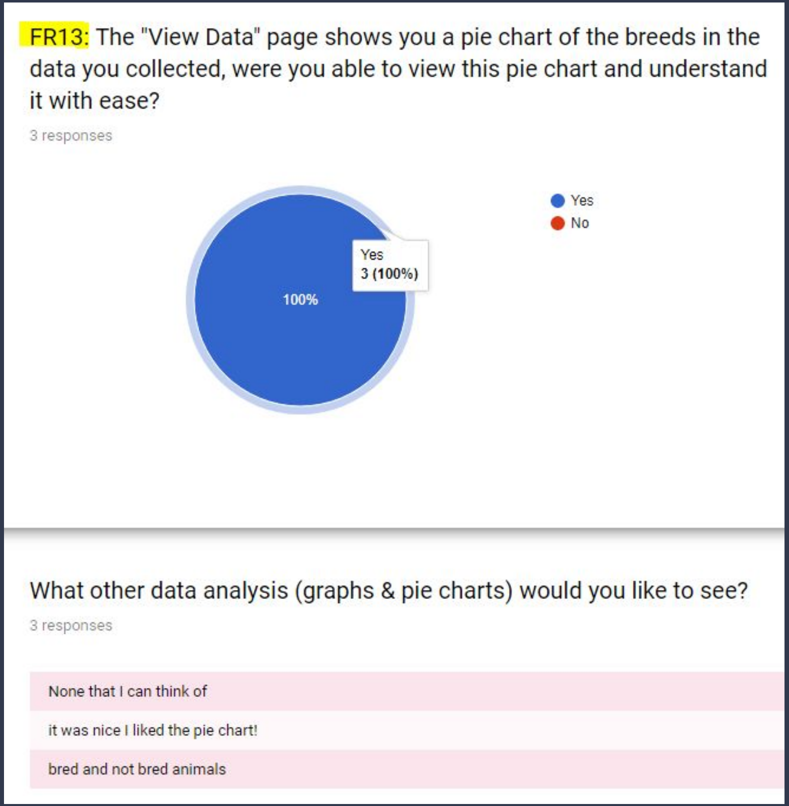
FR5: Were you able to SEARCH through data entries by keywords or drop down menu options?

3 responses

If your answer was "NO" or "other" please explain what the issue was.

1 response

Could not find dropdown



FR13: The "View Data" page shows you a pie chart of the breeds in the data you collected, were you able to view this pie chart and understand it with ease?

3 responses

What other data analysis (graphs & pie charts) would you like to see?

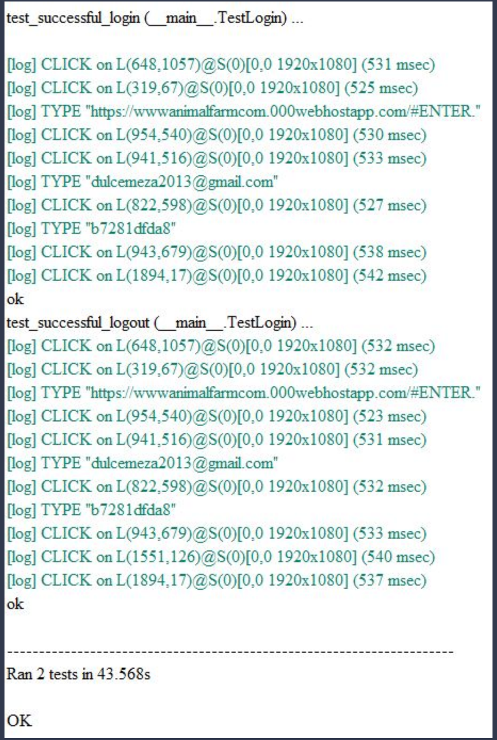
3 responses

None that I can think of

it was nice I liked the pie chart!

bred and not bred animals

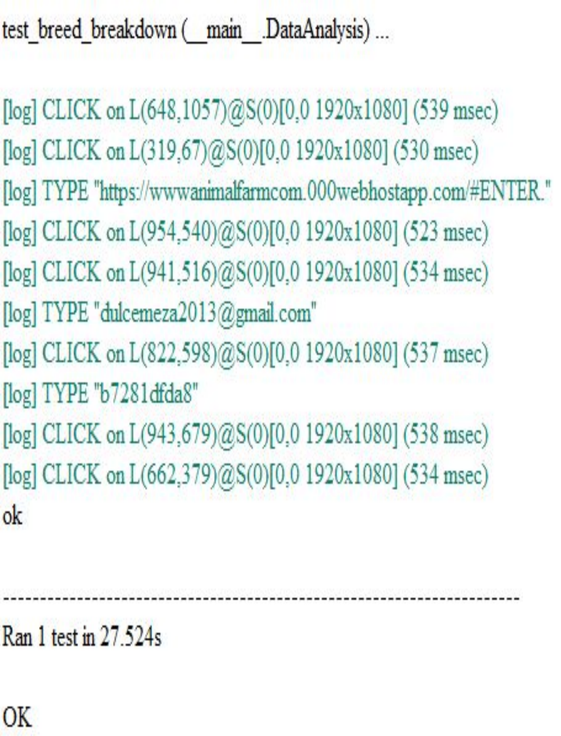
Black-box Testing: Sikuli Test cases (Dulce)



```
test_successful_login (__main__.TestLogin) ...

[log] CLICK on L(648,1057)@S(0)[0,0 1920x1080] (531 msec)
[log] CLICK on L(319,67)@S(0)[0,0 1920x1080] (525 msec)
[log] TYPE "https://wwwanimalfarmcom.000webhostapp.com/#ENTER."
[log] CLICK on L(954,540)@S(0)[0,0 1920x1080] (530 msec)
[log] CLICK on L(941,516)@S(0)[0,0 1920x1080] (533 msec)
[log] TYPE "dulcemeza2013@gmail.com"
[log] CLICK on L(822,598)@S(0)[0,0 1920x1080] (527 msec)
[log] TYPE "b7281dfda8"
[log] CLICK on L(943,679)@S(0)[0,0 1920x1080] (538 msec)
[log] CLICK on L(1894,17)@S(0)[0,0 1920x1080] (542 msec)
ok
test_successful_logout (__main__.TestLogin) ...
[log] CLICK on L(648,1057)@S(0)[0,0 1920x1080] (532 msec)
[log] CLICK on L(319,67)@S(0)[0,0 1920x1080] (532 msec)
[log] TYPE "https://wwwanimalfarmcom.000webhostapp.com/#ENTER."
[log] CLICK on L(954,540)@S(0)[0,0 1920x1080] (523 msec)
[log] CLICK on L(941,516)@S(0)[0,0 1920x1080] (531 msec)
[log] TYPE "dulcemeza2013@gmail.com"
[log] CLICK on L(822,598)@S(0)[0,0 1920x1080] (532 msec)
[log] TYPE "b7281dfda8"
[log] CLICK on L(943,679)@S(0)[0,0 1920x1080] (533 msec)
[log] CLICK on L(1551,126)@S(0)[0,0 1920x1080] (540 msec)
[log] CLICK on L(1894,17)@S(0)[0,0 1920x1080] (537 msec)
ok

----------------------------------------------------------------------
Ran 2 tests in 43.568s

OK
```

- **FR 12**: user shall be able to log in and log out of the website

✓

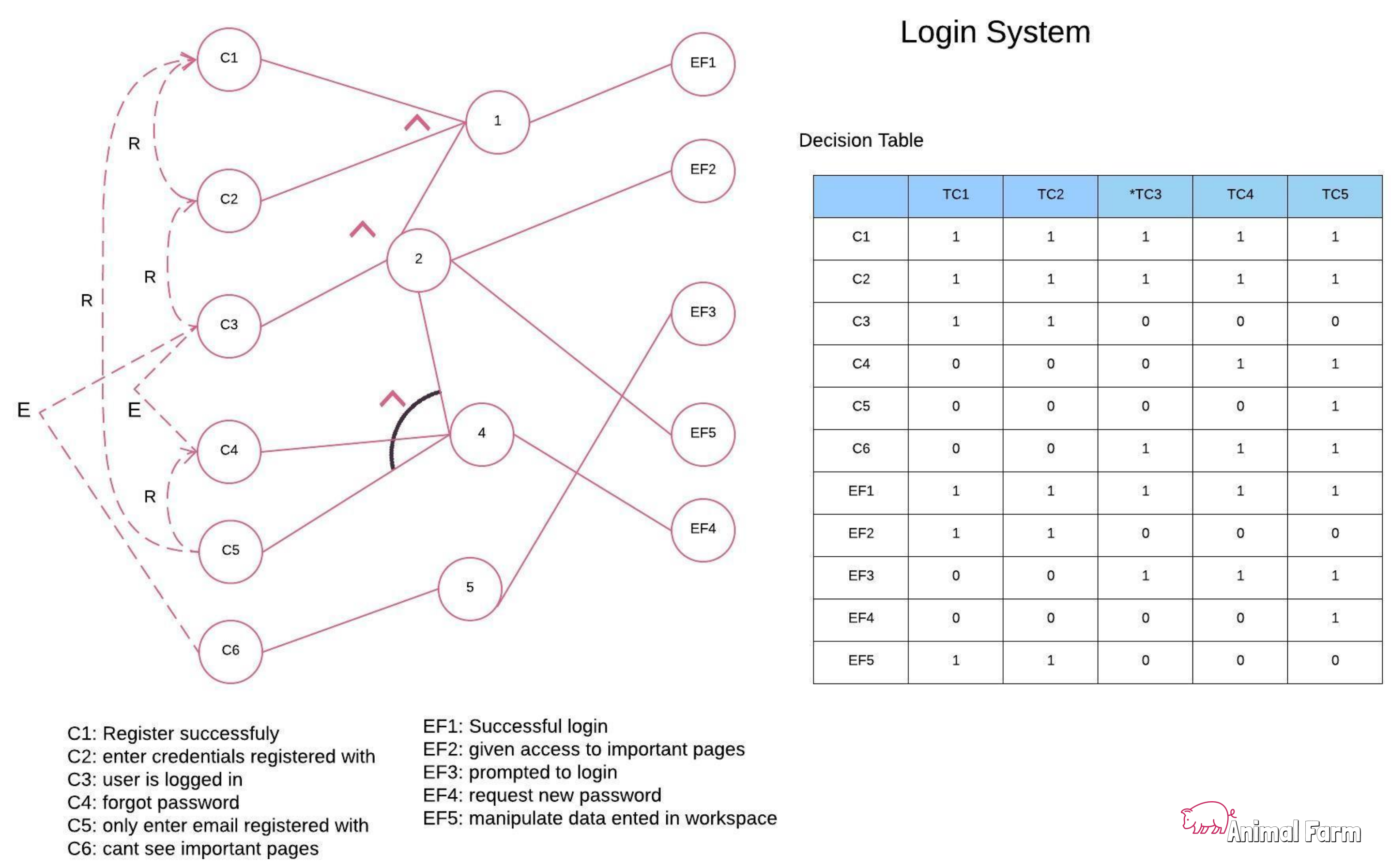


```
test_breed_breakdown (__main__.DataAnalysis) ...

[log] CLICK on L(648,1057)@S(0)[0,0 1920x1080] (539 msec)
[log] CLICK on L(319,67)@S(0)[0,0 1920x1080] (530 msec)
[log] TYPE "https://wwwanimalfarmcom.000webhostapp.com/#ENTER."
[log] CLICK on L(954,540)@S(0)[0,0 1920x1080] (523 msec)
[log] CLICK on L(941,516)@S(0)[0,0 1920x1080] (534 msec)
[log] TYPE "dulcemeza2013@gmail.com"
[log] CLICK on L(822,598)@S(0)[0,0 1920x1080] (537 msec)
[log] TYPE "b7281dfda8"
[log] CLICK on L(943,679)@S(0)[0,0 1920x1080] (538 msec)
[log] CLICK on L(662,379)@S(0)[0,0 1920x1080] (534 msec)
ok

----------------------------------------------------------------------
Ran 1 test in 27.524s

OK
```

- **FR 14**: user shall be able to view a data analysis of the breeds for all data entries

✓

Black-box Testing: Cause-effect Graph (Dulce)



Login System

Decision Table

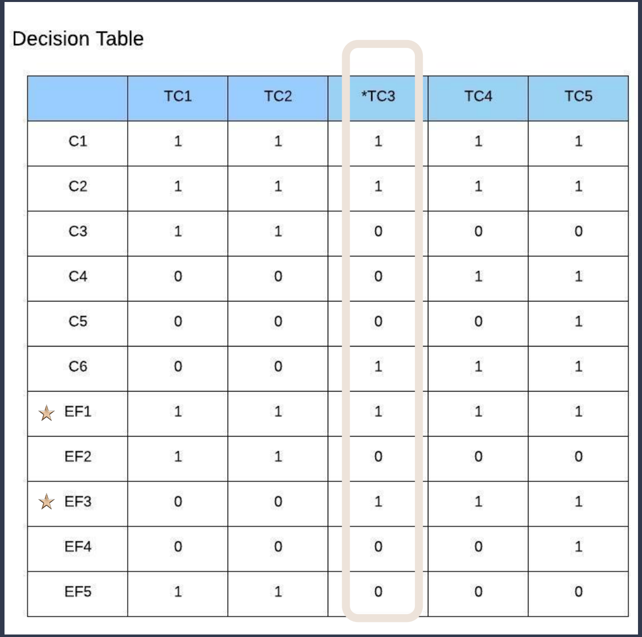
	TC1	TC2	*TC3	TC4	TC5
C1	1	1	1	1	1
C2	1	1	1	1	1
C3	1	1	0	0	0
C4	0	0	0	1	1
C5	0	0	0	0	1
C6	0	0	1	1	1
EF1	1	1	1	1	1
EF2	1	1	0	0	0
EF3	0	0	1	1	1
EF4	0	0	0	0	1
EF5	1	1	0	0	0

C1: Register successfuly
C2: enter credentials registered with
C3: user is logged in
C4: forgot password
C5: only enter email registered with
C6: cant see important pages

EF1: Successful login
EF2: given access to important pages
EF3: prompted to login
EF4: request new password
EF5: manipulate data ented in workspace

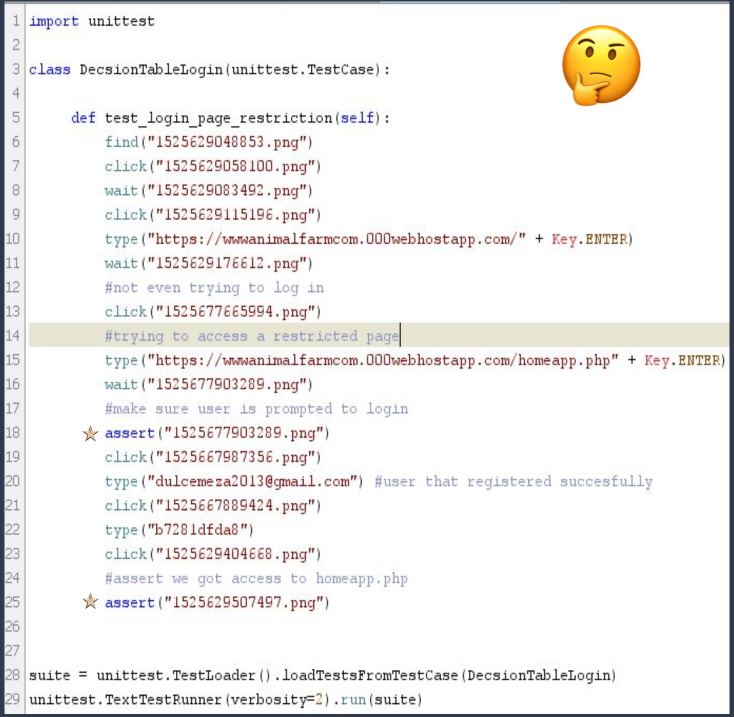
Animal Farm

Black-box Testing: Decision Table (Dulce)

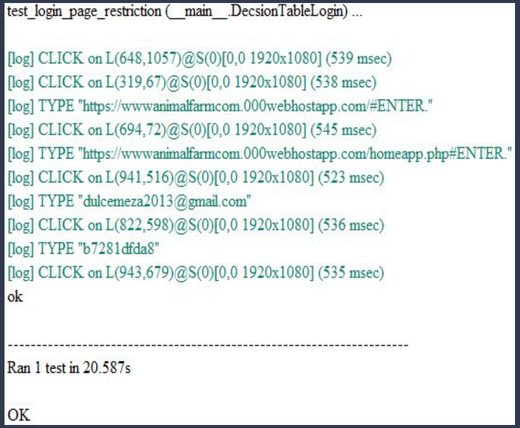


Decision Table

	TC1	TC2	*TC3	TC4	TC5
C1	1	1	1	1	1
C2	1	1	1	1	1
C3	1	1	0	0	0
C4	0	0	0	1	1
C5	0	0	0	0	1
C6	0	0	1	1	1
★ EF1	1	1	1	1	1
EF2	1	1	0	0	0
★ EF3	0	0	1	1	1
EF4	0	0	0	0	1
EF5	1	1	0	0	0



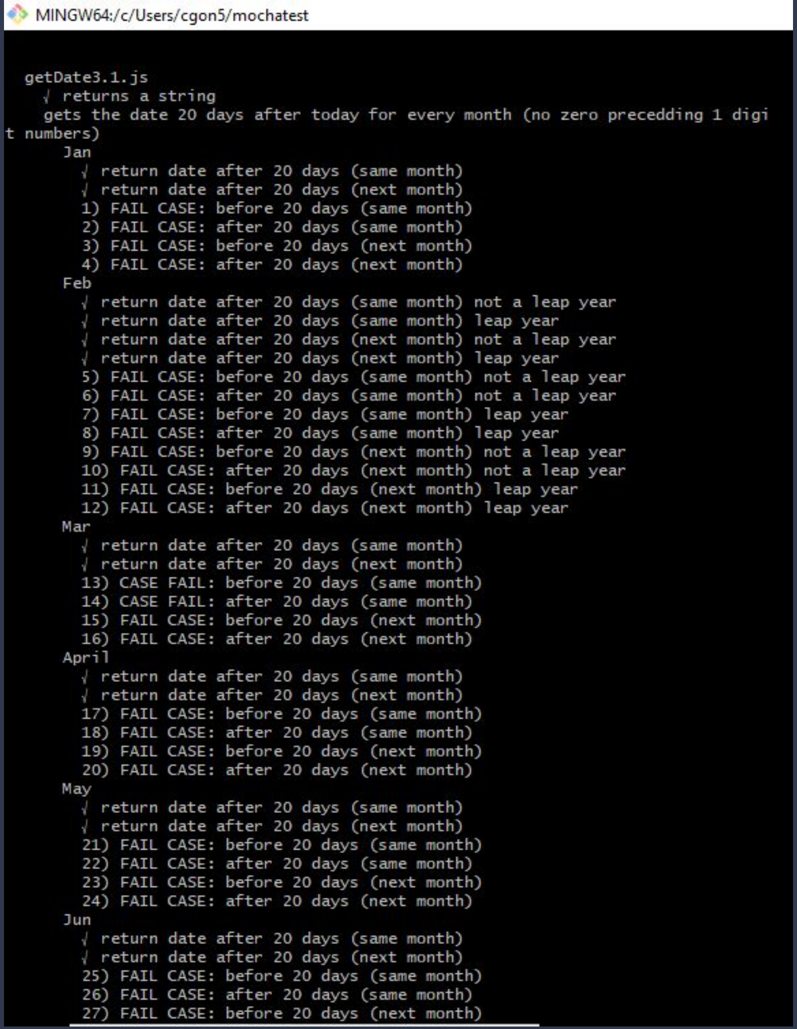
```python
import unittest

class DecsionTableLogin(unittest.TestCase):

    def test_login_page_restriction(self):
        find("1525629048853.png")
        click("1525629058100.png")
        wait("1525629083492.png")
        click("1525629115196.png")
        type("https://wwwanimalfarmcom.000webhostapp.com/" + Key.ENTER)
        wait("1525629176612.png")
        #not even trying to log in
        click("1525677665994.png")
        #trying to access a restricted page
        type("https://wwwanimalfarmcom.000webhostapp.com/homeapp.php" + Key.ENTER)
        wait("1525677903289.png")
        #make sure user is prompted to login
      ★ assert("1525677903289.png")
        click("1525667987356.png")
        type("dulcemeza2013@gmail.com") #user that registered succesfully
        click("1525667889424.png")
        type("b7281dfda8")
        click("1525629404668.png")
        #assert we got access to homeapp.php
      ★ assert("1525629507497.png")


suite = unittest.TestLoader().loadTestsFromTestCase(DecsionTableLogin)
unittest.TextTestRunner(verbosity=2).run(suite)
```



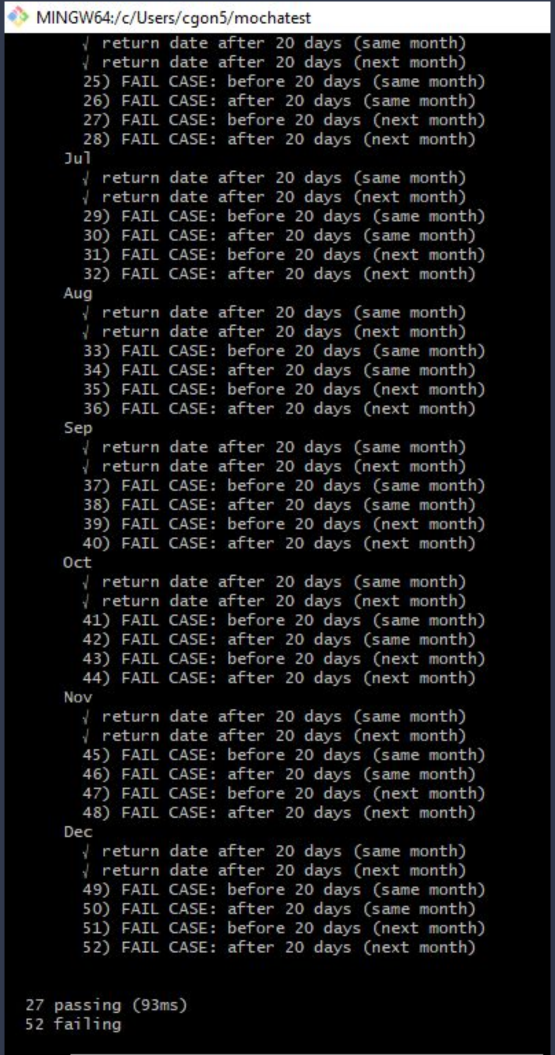
```
test_login_page_restriction (__main__.DecsionTableLogin) ...

[log] CLICK on L(648,1057)@S(0)[0,0 1920x1080] (539 msec)
[log] CLICK on L(319,67)@S(0)[0,0 1920x1080] (538 msec)
[log] TYPE "https://wwwanimalfarmcom.000webhostapp.com/#ENTER."
[log] CLICK on L(694,72)@S(0)[0,0 1920x1080] (545 msec)
[log] TYPE "https://wwwanimalfarmcom.000webhostapp.com/homeapp.php#ENTER."
[log] CLICK on L(941,516)@S(0)[0,0 1920x1080] (523 msec)
[log] TYPE "dulcemeza2013@gmail.com"
[log] CLICK on L(822,598)@S(0)[0,0 1920x1080] (536 msec)
[log] TYPE "b7281dfda8"
[log] CLICK on L(943,679)@S(0)[0,0 1920x1080] (535 msec)
ok

----------------------------------------------------------------------
Ran 1 test in 20.587s

OK
```

White-Box Testing

White-box Testing (Chris)



```
  getDate3.1.js
    √ returns a string
    gets the date 20 days after today for every month (no zero precedding 1 digi
t numbers)
      Jan
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        1) FAIL CASE: before 20 days (same month)
        2) FAIL CASE: after 20 days (same month)
        3) FAIL CASE: before 20 days (next month)
        4) FAIL CASE: after 20 days (next month)
      Feb
        √ return date after 20 days (same month) not a leap year
        √ return date after 20 days (same month) leap year
        √ return date after 20 days (next month) not a leap year
        √ return date after 20 days (next month) leap year
        5) FAIL CASE: before 20 days (same month) not a leap year
        6) FAIL CASE: after 20 days (same month) not a leap year
        7) FAIL CASE: before 20 days (same month) leap year
        8) FAIL CASE: after 20 days (same month) leap year
        9) FAIL CASE: before 20 days (next month) not a leap year
        10) FAIL CASE: after 20 days (next month) not a leap year
        11) FAIL CASE: before 20 days (next month) leap year
        12) FAIL CASE: after 20 days (next month) leap year
      Mar
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        13) CASE FAIL: before 20 days (same month)
        14) CASE FAIL: after 20 days (same month)
        15) FAIL CASE: before 20 days (next month)
        16) FAIL CASE: after 20 days (next month)
      April
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        17) FAIL CASE: before 20 days (same month)
        18) FAIL CASE: after 20 days (same month)
        19) FAIL CASE: before 20 days (next month)
        20) FAIL CASE: after 20 days (next month)
      May
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        21) FAIL CASE: before 20 days (same month)
        22) FAIL CASE: after 20 days (same month)
        23) FAIL CASE: before 20 days (next month)
        24) FAIL CASE: after 20 days (next month)
      Jun
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        25) FAIL CASE: before 20 days (same month)
        26) FAIL CASE: after 20 days (same month)
        27) FAIL CASE: before 20 days (next month)
```



```
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        25) FAIL CASE: before 20 days (same month)
        26) FAIL CASE: after 20 days (same month)
        27) FAIL CASE: before 20 days (next month)
        28) FAIL CASE: after 20 days (next month)
      Jul
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        29) FAIL CASE: before 20 days (same month)
        30) FAIL CASE: after 20 days (same month)
        31) FAIL CASE: before 20 days (next month)
        32) FAIL CASE: after 20 days (next month)
      Aug
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        33) FAIL CASE: before 20 days (same month)
        34) FAIL CASE: after 20 days (same month)
        35) FAIL CASE: before 20 days (next month)
        36) FAIL CASE: after 20 days (next month)
      Sep
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        37) FAIL CASE: before 20 days (same month)
        38) FAIL CASE: after 20 days (same month)
        39) FAIL CASE: before 20 days (next month)
        40) FAIL CASE: after 20 days (next month)
      Oct
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        41) FAIL CASE: before 20 days (same month)
        42) FAIL CASE: after 20 days (same month)
        43) FAIL CASE: before 20 days (next month)
        44) FAIL CASE: after 20 days (next month)
      Nov
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        45) FAIL CASE: before 20 days (same month)
        46) FAIL CASE: after 20 days (same month)
        47) FAIL CASE: before 20 days (next month)
        48) FAIL CASE: after 20 days (next month)
      Dec
        √ return date after 20 days (same month)
        √ return date after 20 days (next month)
        49) FAIL CASE: before 20 days (same month)
        50) FAIL CASE: after 20 days (same month)
        51) FAIL CASE: before 20 days (next month)
        52) FAIL CASE: after 20 days (next month)


  27 passing (93ms)
  52 failing
```

Demo

Thank you!

:)