Testing Document
Team 3

Sara Wiltberger 5508442445
Cole Brossart 8018427432
Adrian Sunga 4355385501
Elizabeth Worstell 7931092914
Natalie Mackraz 2414000712

Table of Contents

Preface

This document contains all the information regarding testing our ParkHere application. This documentation first gives instructions on how to download and configure the testing frameworks we used (Protractor for blackbox, and Karma and Jasmine for whitebox tests). Next, we document the location, description, execution instructions, input and rational, and results of our black box testing. Lastly, we do the same for whitebox testing.

Instructions

Black Box Framework Download Instructions

1. We used the Protractor framework to run our black box tests. First protracts through npm:

```
$ npm install -g protractor
$ webdriver-manager update
```

2. Start the ionic browser:

```
ionic serve --nobrowser
```

3. In a new terminal tab, cd to ParkHere/myApp/ionic-testing/
4. Run the commands for each test. Instructions are found below under Black Box Tests

White Box Framework Download Instructions

1. Insall Jasmine and Karma:

```
$ npm install karma --save-dev
$ npm install karma-jasmine --save-dev
$ bower install angular-mocks#1.3.13 --save-dev
```

2. Install the karma command line interface and ensure you have all the packages:

```
$ npm install -g karma-cli
```

3. Install the phantom browser

```
$ npm install -g phantomjs
```

4. Run the tests as described below, or run all tests by cd into the tests folder and:

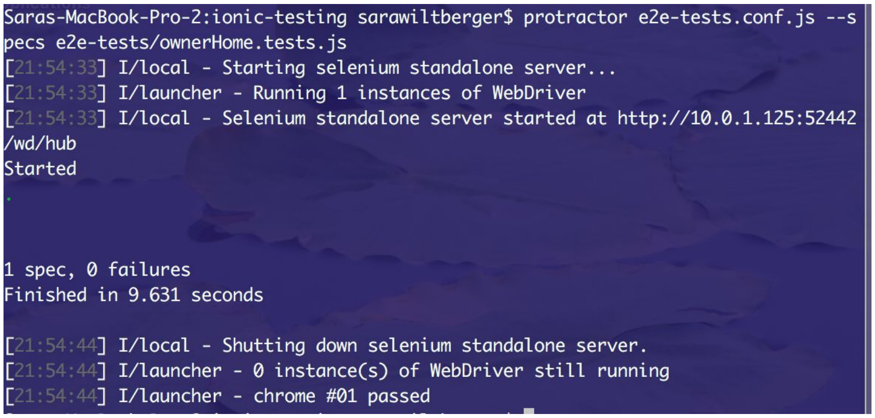
```
karma start unit-test.conf.js
```

Blackbox Tests

The black box tests were performed using Protractor framework. Information on how to download can be found at http://www.protractortest.org (or see above instructions). Protractor runs the front end view and allows automated user input. The ionic framework we used allows me to view the android app in a webpage. I tested on this because it was quicker, so all my screenshots are from that. Also, inserting data into popups was not supported by this framework (since the popups were an ionic plug in and not native to ionic). For all testing involving popups, I had to pause the framework and input the info manually. I understand this is less than ideal, but no other framework gave me this functionality.

Sign Up

Test 1: Successful Owner Signup

- Location: ParkHere/myApp/ionic-testing/e2e-tests/signup.tests.js
    - under it function 'owner successful sign up'
- Description: Using an email that is not already used in ParkHere, I sign up a new owner with email, name, password
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/signup.tests.js
        - Note: this will run all signup tests
- Rationale: I chose the inputs (email, password, click owner) of a user I knew was not signed up. I wanted to ensure that I was able to add new users to the app. After I added these users to the database, I checked that I could log into their accounts, (see below).
- Results:
- 

```
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --s
pecs e2e-tests/ownerHome.tests.js
[21:54:33] I/local - Starting selenium standalone server...
[21:54:33] I/launcher - Running 1 instances of WebDriver
[21:54:33] I/local - Selenium standalone server started at http://10.0.1.125:52442
/wd/hub
Started
.


1 spec, 0 failures
Finished in 9.631 seconds

[21:54:44] I/local - Shutting down selenium standalone server.
[21:54:44] I/launcher - 0 instance(s) of WebDriver still running
[21:54:44] I/launcher - chrome #01 passed
```

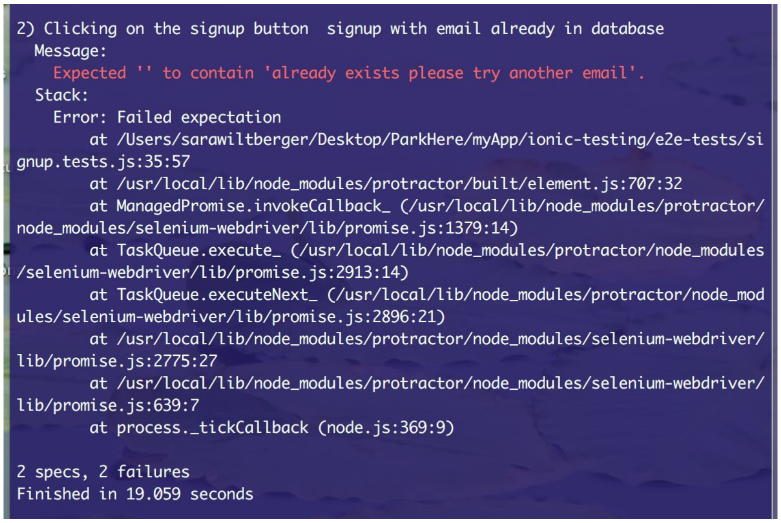
    - The test passed with no failures. What can't be captured in pictures is the front end view being automated

    - Note: if you run these tests currently, they will fail because I ran the tests already and signed up these users into ParkHere (thus the email is already used).

Test 2: Successful Parker Sign up

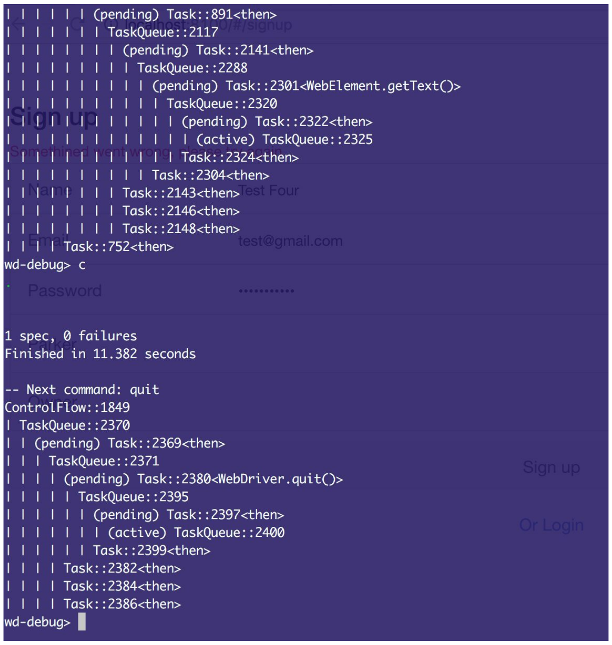
- This test is exactly the same as above (also ran the same because the test is in the same file, and the results are on the screenshot) but with clicking parker instead. It worked.

Test 3: Sign up when user already in database

- Location: ParkHere/myApp/ionic-testing/e2e-tests/signup.tests.js
    - under it function 'signup with email already in database'
- Description: Signing up with an email already being use should produce a warning error
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/signup.tests.js
        - Note: this will run all signup tests
- Rationale: We need to let the user know that her sign up failed and she needs to try a different emails
- Results:
- 

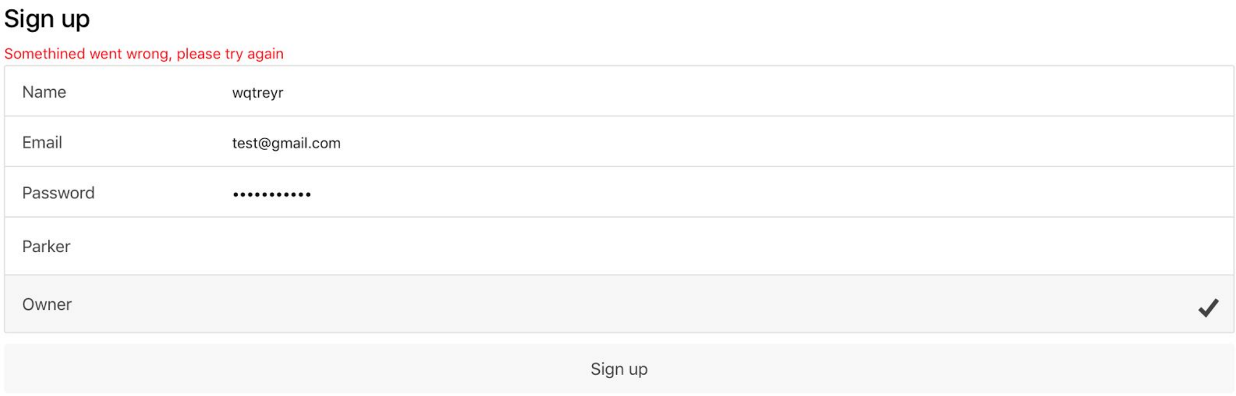
```
2) Clicking on the signup button  signup with email already in database
  Message:
    Expected '' to contain 'already exists please try another email'.
  Stack:
    Error: Failed expectation
        at /Users/sarawiltberger/Desktop/ParkHere/myApp/ionic-testing/e2e-tests/si
gnup.tests.js:35:57
        at /usr/local/lib/node_modules/protractor/built/element.js:707:32
        at ManagedPromise.invokeCallback_ (/usr/local/lib/node_modules/protractor/
node_modules/selenium-webdriver/lib/promise.js:1379:14)
        at TaskQueue.execute_ (/usr/local/lib/node_modules/protractor/node_modules
/selenium-webdriver/lib/promise.js:2913:14)
        at TaskQueue.executeNext_ (/usr/local/lib/node_modules/protractor/node_mod
ules/selenium-webdriver/lib/promise.js:2896:21)
        at /usr/local/lib/node_modules/protractor/node_modules/selenium-webdriver/
lib/promise.js:2775:27
        at /usr/local/lib/node_modules/protractor/node_modules/selenium-webdriver/
lib/promise.js:639:7
        at process._tickCallback (node.js:369:9)

2 specs, 2 failures
Finished in 19.059 seconds
```

    - The test failed the first time. The front end view was not producing the warning.
    - I fixed this by going to the controllers.js signUpCtrl and fixing the error that is caught to produce the message

- 

```
| | | | | | (pending) Task::891<then>
| | | | | | | TaskQueue::2117
| | | | | | | | (pending) Task::2141<then>
| | | | | | | | | TaskQueue::2288
| | | | | | | | | | (pending) Task::2301<WebElement.getText()>
| | | | | | | | | | | TaskQueue::2320
| | | | | | | | | | | | (pending) Task::2322<then>
| | | | | | | | | | | | | (active) TaskQueue::2325
| | | | | | | | | | | | Task::2324<then>
| | | | | | | | | | Task::2304<then>
| | | | | | | | Task::2143<then>
| | | | | | | | Task::2146<then>
| | | | | | | | Task::2148<then>
| | | | Task::752<then>
wd-debug> c


1 spec, 0 failures
Finished in 11.382 seconds

-- Next command: quit
ControlFlow::1849
| TaskQueue::2370
| | (pending) Task::2369<then>
| | | TaskQueue::2371
| | | | (pending) Task::2380<WebDriver.quit()>
| | | | | TaskQueue::2395
| | | | | | (pending) Task::2397<then>
| | | | | | | (active) TaskQueue::2400
| | | | | | Task::2399<then>
| | | | Task::2382<then>
| | | | Task::2384<then>
| | | | Task::2386<then>
wd-debug>
```

- This shows no errors in debug mode

- 

Sign up

Somethined went wrong, please try again

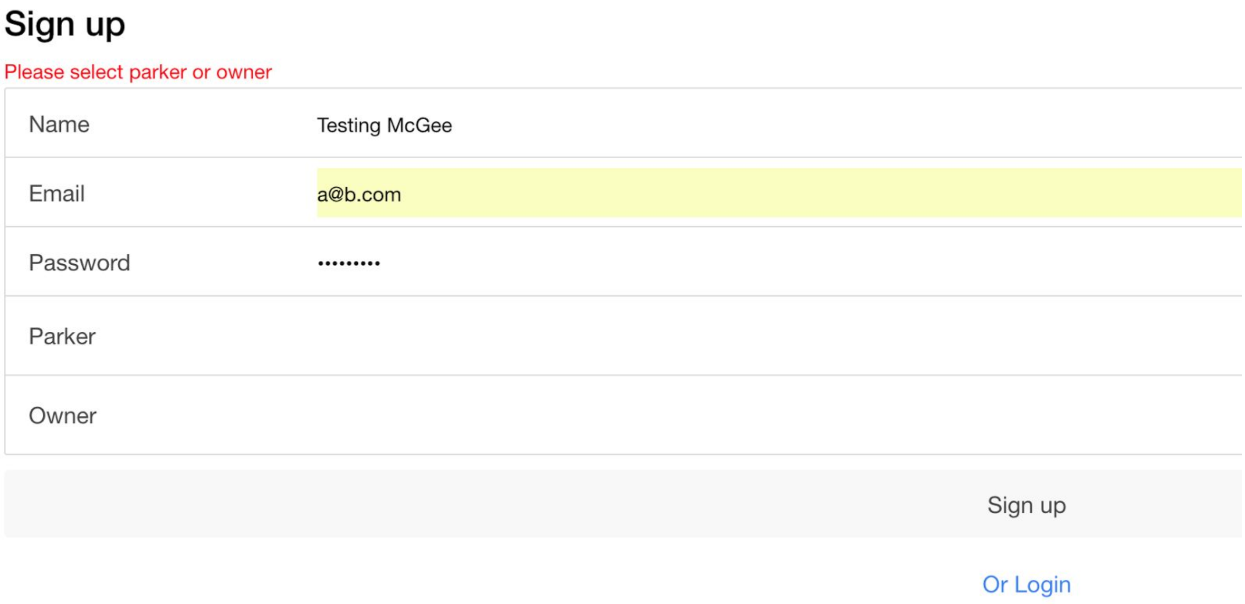
Name	wqtreyr
Email	test@gmail.com
Password	•••••••••••
Parker	
Owner	✓

- Here is the front end with the error message

Test 4: Signup without Owner/Parker Selected Fails

- Location: ParkHere/myApp/ionic-testing/e2e-tests/signup.tests.js
    - under it function 'should fail without a usertype selected'
- Description: Signing up without selecting either the Parker or Owner causes an error warning
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/signup.tests.js
        - Note: this will run all signup tests
- Rationale: We need to send a warning message so that the user selects a usertype so she can select one and proceed.

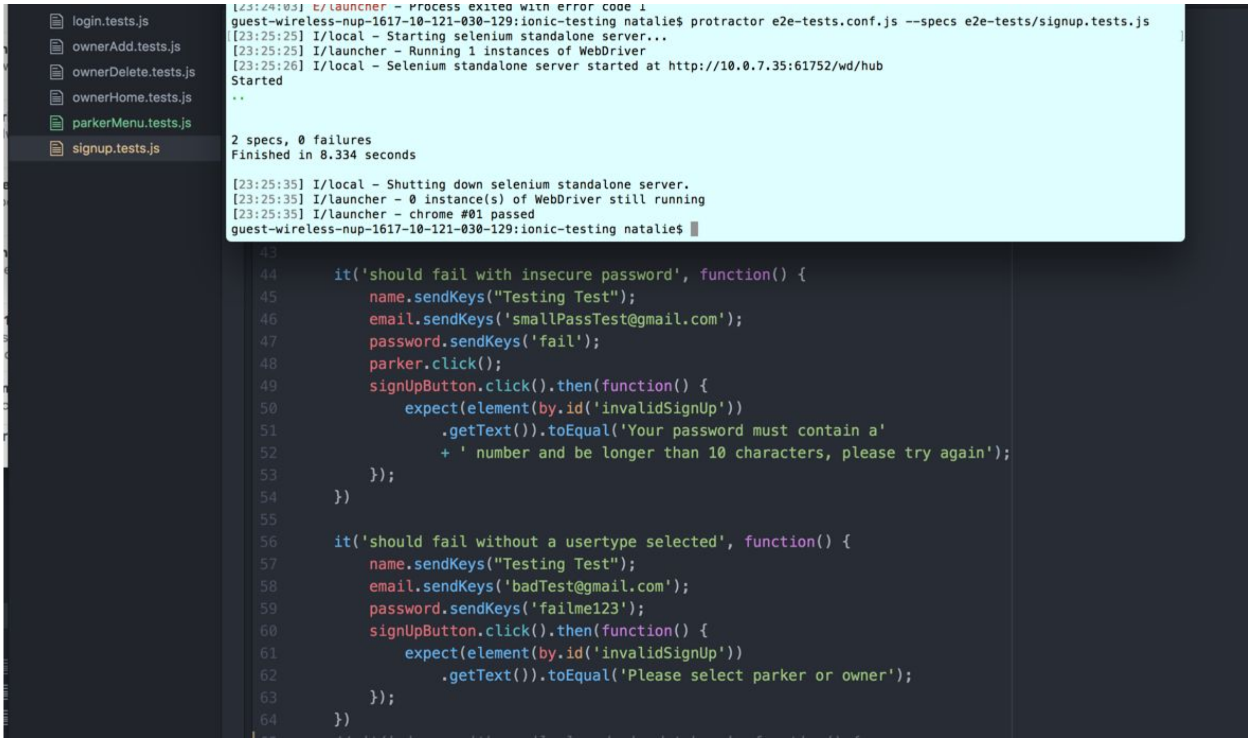
- Results:



Sign up

Please select parker or owner

Name	Testing McGee
Email	a@b.com
Password	•••••••••
Parker	
Owner	

  - Here is the front-end. This test passes with no failures.
  - 

```
[23:24:03] E/launcher - Process exited with error code 1
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$ protractor e2e-tests.conf.js --specs e2e-tests/signup.tests.js
[23:25:25] I/local - Starting selenium standalone server...
[23:25:25] I/launcher - Running 1 instances of WebDriver
[23:25:26] I/local - Selenium standalone server started at http://10.0.7.35:61752/wd/hub
Started
..


2 specs, 0 failures
Finished in 8.334 seconds

[23:25:35] I/local - Shutting down selenium standalone server.
[23:25:35] I/launcher - 0 instance(s) of WebDriver still running
[23:25:35] I/launcher - chrome #01 passed
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$
```

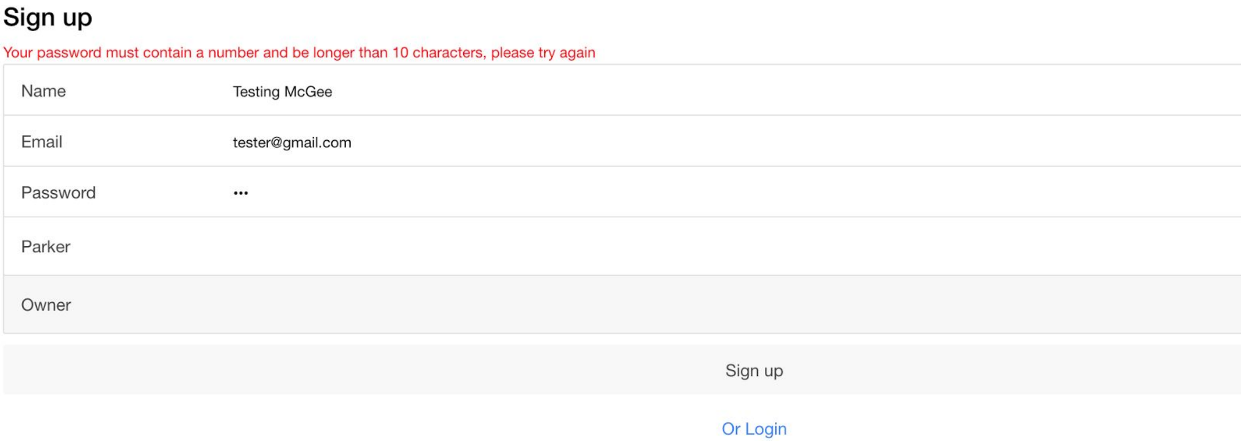
```

    it('should fail with insecure password', function() {
        name.sendKeys("Testing Test");
        email.sendKeys('smallPassTest@gmail.com');
        password.sendKeys('fail');
        parker.click();
        signUpButton.click().then(function() {
            expect(element(by.id('invalidSignUp'))
                .getText()).toEqual('Your password must contain a'
                + ' number and be longer than 10 characters, please try again');
        });
    })

    it('should fail without a usertype selected', function() {
        name.sendKeys("Testing Test");
        email.sendKeys('badTest@gmail.com');
        password.sendKeys('failme123');
        signUpButton.click().then(function() {
            expect(element(by.id('invalidSignUp'))
                .getText()).toEqual('Please select parker or owner');
        });
    })
```

Test 5: Signup with Invalid Password Fails

- Location: ParkHere/myApp/ionic-testing/e2e-tests/signup.tests.js
  - under it function 'should fail with insecure password'
- Description: Signing up with an insecure password causes an error warning
- Execution:

    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/signup.tests.js
        - Note: this will run all signup tests

- Rationale: We need to send a warning message so that the user provides a secure enough password in order to lessen their chances of being hacked in the future so they have to correct it before proceeding.
- Results:



Sign up

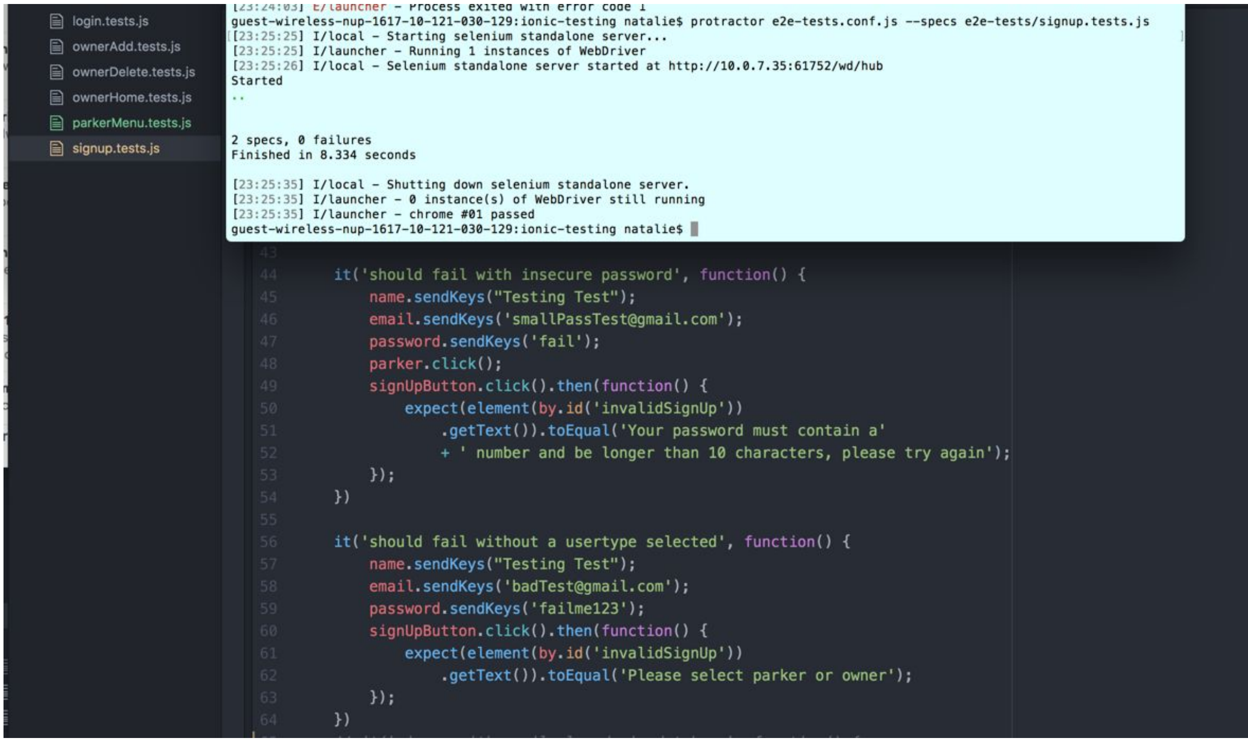
Your password must contain a number and be longer than 10 characters, please try again

Name Testing McGee

Email tester@gmail.com

Password •••

    - Here is the front-end once the bug is fixed (before it was the same screen but with no message viewable). All tests pass.
    -



```
[23:24:03] E/launcher - Process exited with error code 1
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$ protractor e2e-tests.conf.js --specs e2e-tests/signup.tests.js
[23:25:25] I/local - Starting selenium standalone server...
[23:25:25] I/launcher - Running 1 instances of WebDriver
[23:25:26] I/local - Selenium standalone server started at http://10.0.7.35:61752/wd/hub
Started
..


2 specs, 0 failures
Finished in 8.334 seconds

[23:25:35] I/local - Shutting down selenium standalone server.
[23:25:35] I/launcher - 0 instance(s) of WebDriver still running
[23:25:35] I/launcher - chrome #01 passed
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$
```

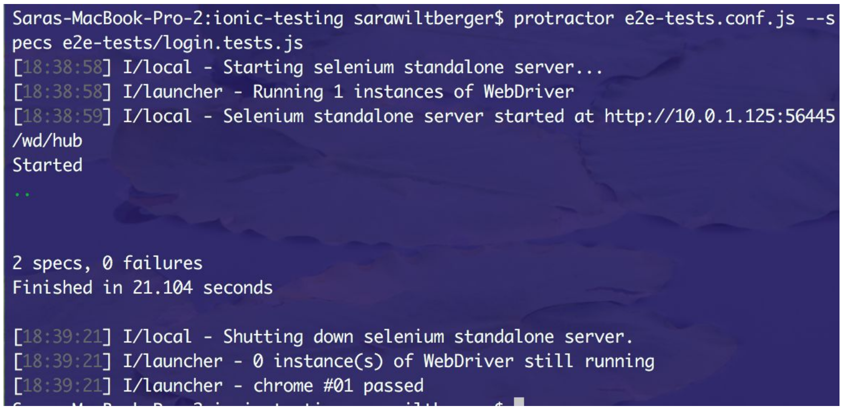
```
    it('should fail with insecure password', function() {
        name.sendKeys("Testing Test");
        email.sendKeys('smallPassTest@gmail.com');
        password.sendKeys('fail');
        parker.click();
        signUpButton.click().then(function() {
            expect(element(by.id('invalidSignUp'))
                .getText()).toEqual('Your password must contain a'
                + ' number and be longer than 10 characters, please try again');
        });
    })

    it('should fail without a usertype selected', function() {
        name.sendKeys("Testing Test");
        email.sendKeys('badTest@gmail.com');
        password.sendKeys('failme123');
        signUpButton.click().then(function() {
            expect(element(by.id('invalidSignUp'))
                .getText()).toEqual('Please select parker or owner');
        });
    })
```

- Bug Fix: Through this test we discovered that the the warning was appearing on the login page instead of the sign up page. This required a change to the html tag id on the sign up page since they were identical. Once this change was made the test passed as expected.

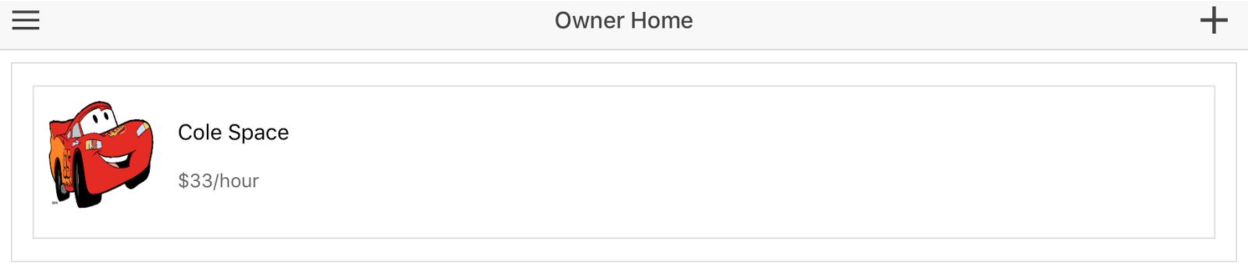
Login:

Test 1: Successful Owner Login

- Location: ParkHere/myApp/ionic-testing/e2e-tests/login.tests.js
    - under it function 'with OWNER should validate the credentials for a successful login and display owner home'
- Description: Given a signed up owner, logging in will direct them to the owner homepage with showing the spaces they have added. I test that they are properly redirected to the owner home page by checking the url.
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/login.tests.js
        - Note: this will run all login tests
- Rationale: I chose the inputs (email, password, click owner) of a signed up owner with spaces saved to ensure I can login as this owner. The test is generated by automating the login process. I also tried with multiple different signed up users, but the current code just shows one login example.
- Results:



```
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --s
pecs e2e-tests/login.tests.js
[18:38:58] I/local - Starting selenium standalone server...
[18:38:58] I/launcher - Running 1 instances of WebDriver
[18:38:59] I/local - Selenium standalone server started at http://10.0.1.125:56445
/wd/hub
Started
..


2 specs, 0 failures
Finished in 21.104 seconds

[18:39:21] I/local - Shutting down selenium standalone server.
[18:39:21] I/launcher - 0 instance(s) of WebDriver still running
[18:39:21] I/launcher - chrome #01 passed
```

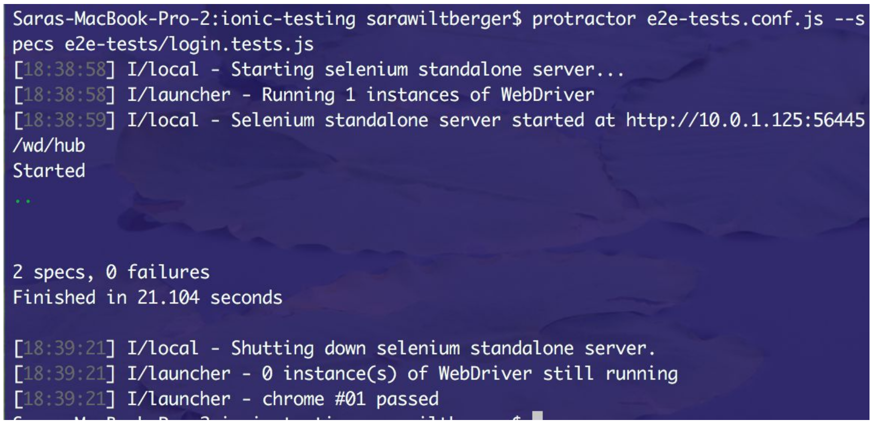
- 

-

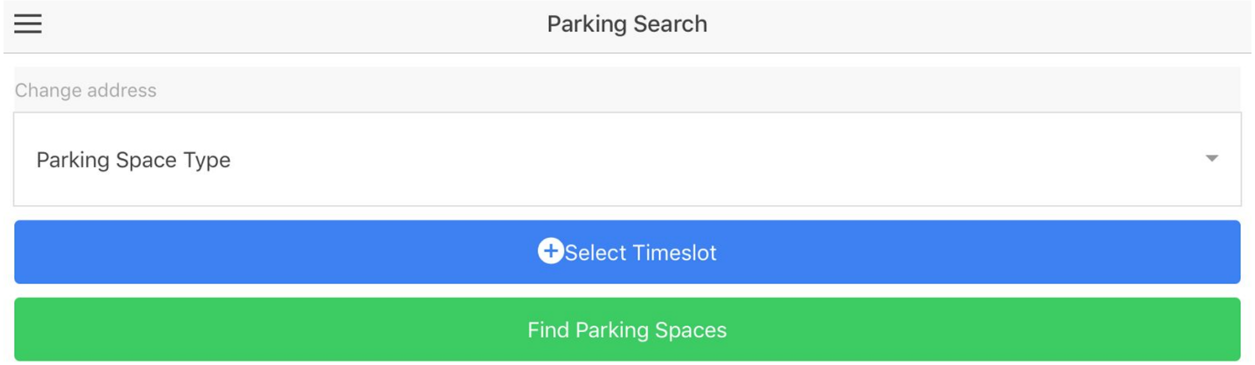
    - The test passed with no failures. The owner was redirected to owner home. What can't be captured in pictures is the front end view being automated
- This function works as expected

Test 2: Parker Login

- Location: ParkHere/myApp/ionic-testing/e2e-tests/login.tests.js
    - under it function 'with PARKER should validate the credentials for a successful login and display parker search'
- Description: Given a signed up parker, logging in will direct them to the parker search page with showing the spaces they have added. I test that they are properly redirected to the parker page by checking the url.
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/login.tests.js
        - Note: this will run all login tests
- Rationale: I chose the inputs (email, password, click owner) of a signed up parker. The test is generated by automating the login process. I also tried with multiple different signed up users, but the current code just shows one login example.
- Results:



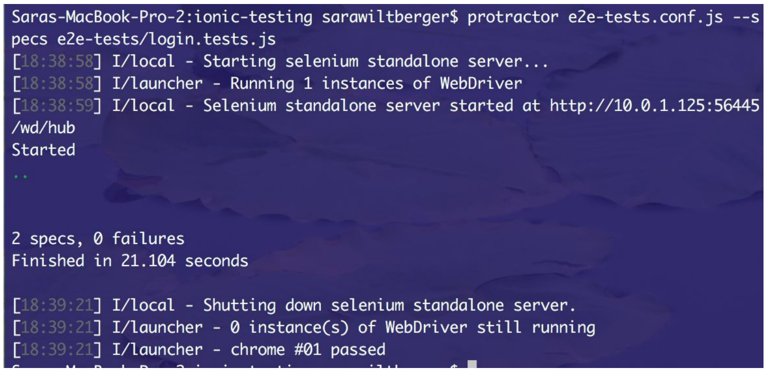
```
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --s
pecs e2e-tests/login.tests.js
[18:38:58] I/local - Starting selenium standalone server...
[18:38:58] I/launcher - Running 1 instances of WebDriver
[18:38:59] I/local - Selenium standalone server started at http://10.0.1.125:56445
/wd/hub
Started
..


2 specs, 0 failures
Finished in 21.104 seconds

[18:39:21] I/local - Shutting down selenium standalone server.
[18:39:21] I/launcher - 0 instance(s) of WebDriver still running
[18:39:21] I/launcher - chrome #01 passed
```

- 

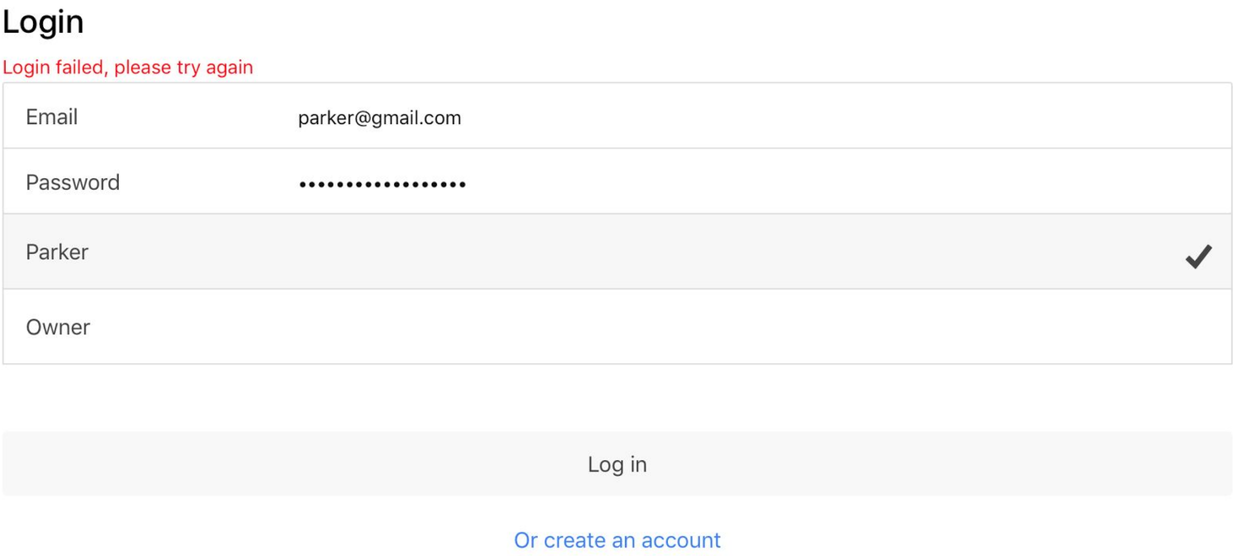
-
    - The test passed with no failures. I was redirected to the parker search

- This function works as expected

Test 3: Unsuccessful Login

- Location: ParkHere/myApp/ionic-testing/e2e-tests/login.tests.js
  - under it function 'should display an error for an unsuccessful login'
- Description: Given a user that is not signed up, the login page will will produce an error and let them resign up
- Execution:
  - from ParkHere/myApp/ionic-testing
  - run: protractor e2e-tests.conf.js --specs e2e-tests/login.tests.js
    - Note: this will run all login tests
- Rationale: I chose the inputs (email, password, click owner) of a user I knew was not signed up. I then tried to login with a user I knew did exist
- Results:
- 

```
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --s
pecs e2e-tests/login.tests.js
[18:38:58] I/local - Starting selenium standalone server...
[18:38:58] I/launcher - Running 1 instances of WebDriver
[18:38:59] I/local - Selenium standalone server started at http://10.0.1.125:56445
/wd/hub
Started
..


2 specs, 0 failures
Finished in 21.104 seconds

[18:39:21] I/local - Shutting down selenium standalone server.
[18:39:21] I/launcher - 0 instance(s) of WebDriver still running
[18:39:21] I/launcher - chrome #01 passed
```



Login

Login failed, please try again

Email parker@gmail.com

Password ••••••••••••••••••

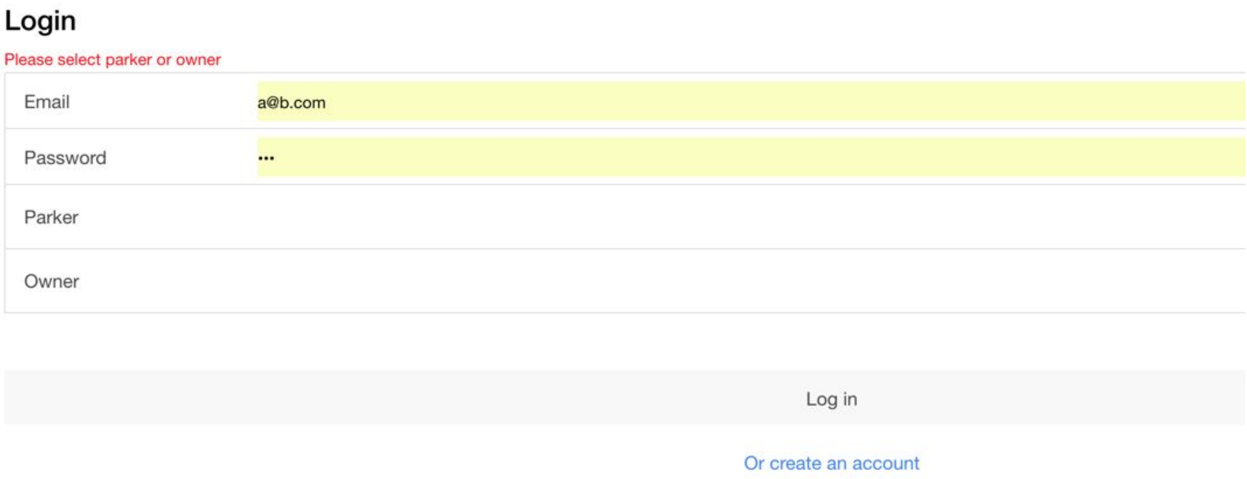
Parker ✓

Owner

-
  - The test passed with no failures. An error message appeared when I inputted the wrong password

Test 4: Login without Owner/Parker Selected Fails

- Location: ParkHere/myApp/ionic-testing/e2e-tests/signup.tests.js
    - under it function 'should fail without a usertype selected'
- Description: Signing up without selecting either the Parker or Owner causes an error warning
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/signup.tests.js
        - Note: this will run all signup tests
- Rationale: The email and password selected were ones that we knew worked when parker was selected. We need to send a warning message so that the user selects which type (parker or owner) they want to log in as (as some users may have an account for each so we need to know which to display).
- Results:



Login

Please select parker or owner

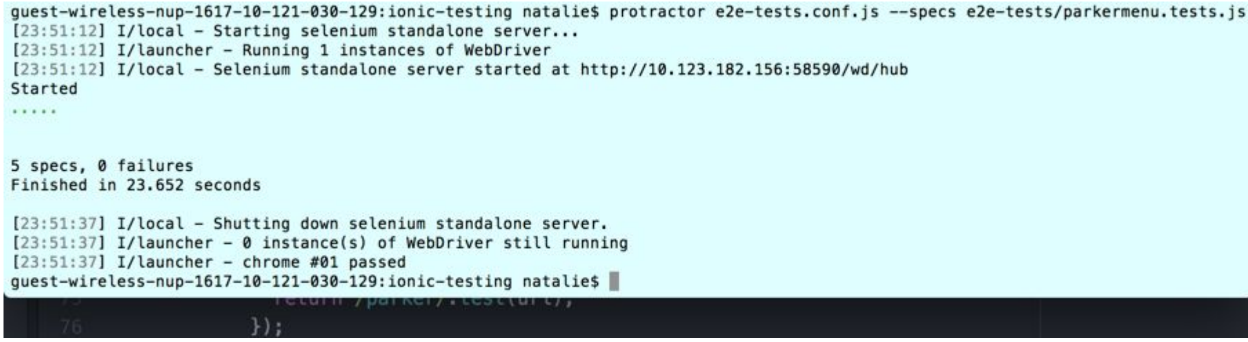
Email	a@b.com
Password	•••
Parker	
Owner	

    - Front end of test. All tests path

Parker Menu Test

Test 1: Parker Menu option Upcoming Spaces leads to the upcoming spaces page

- Location: ParkHere/myApp/ionic-testing/e2e-tests/parkerMenu.tests.js
    - under it function 'parker menu Upcoming Spaces --> upcoming spaces page'
- Description: When a signed in parker opens the app and they toggle open the menu, clicking on the Upcoming Spaces option should lead them to the page that displays their upcoming spaces.
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/parkerMenu.tests.js

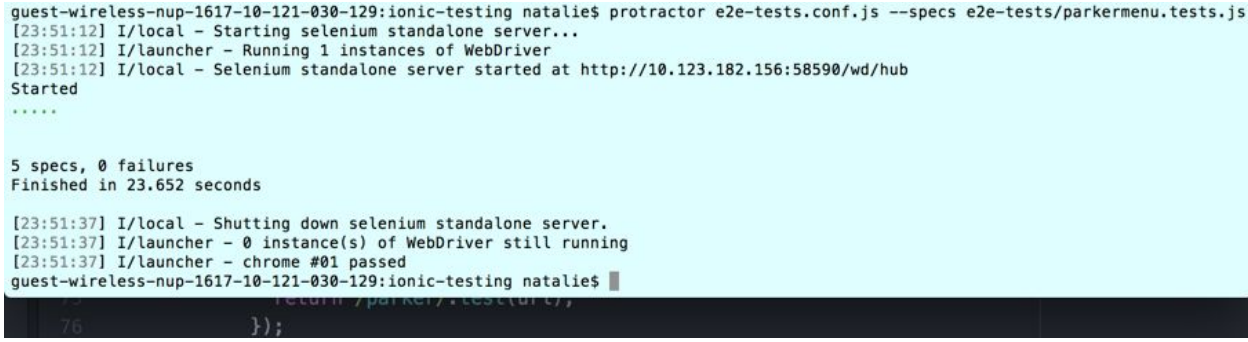
- Rationale: I correctly logged in under a parker account that I knew would work properly. From there I opened the menu and selecting upcoming spaces and waited for the page to load (giving it a 5 second window to ensure that it was not too slow although every time it has been run so far it's been almost instantaneous).
- Results:
  - Since this test requires much automation, it’s not viewable via screenshot.
  - 

```
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$ protractor e2e-tests.conf.js --specs e2e-tests/parkermenu.tests.js
[23:51:12] I/local - Starting selenium standalone server...
[23:51:12] I/launcher - Running 1 instances of WebDriver
[23:51:12] I/local - Selenium standalone server started at http://10.123.182.156:58590/wd/hub
Started
.....


5 specs, 0 failures
Finished in 23.652 seconds

[23:51:37] I/local - Shutting down selenium standalone server.
[23:51:37] I/launcher - 0 instance(s) of WebDriver still running
[23:51:37] I/launcher - chrome #01 passed
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$
```

```
            });
```

    - All parkerMenu tests pass

Test 2: Parker Menu option Map View leads to the map view page

- Location: ParkHere/myApp/ionic-testing/e2e-tests/parkerMenu.tests.js
  - under it function 'parker menu Map View --> mapView page'
- Description: When a signed in parker opens the app and they toggle open the menu, clicking on the Map View option should lead them to the page that correctly displays a map view of spots around them.
- Execution:
  - from ParkHere/myApp/ionic-testing
  - run: protractor e2e-tests.conf.js --specs e2e-tests/parkerMenu.tests.js
- Rationale: I correctly logged in under a parker account that I knew would work properly. From there I opened the menu and selecting map view and waited for the page to load (giving it a 5 second window to ensure that it was not too slow although every time it has been run so far it's been almost instantaneous).
- Results:
  - Since this test requires much automation, it’s not viewable via screenshot.
  - 

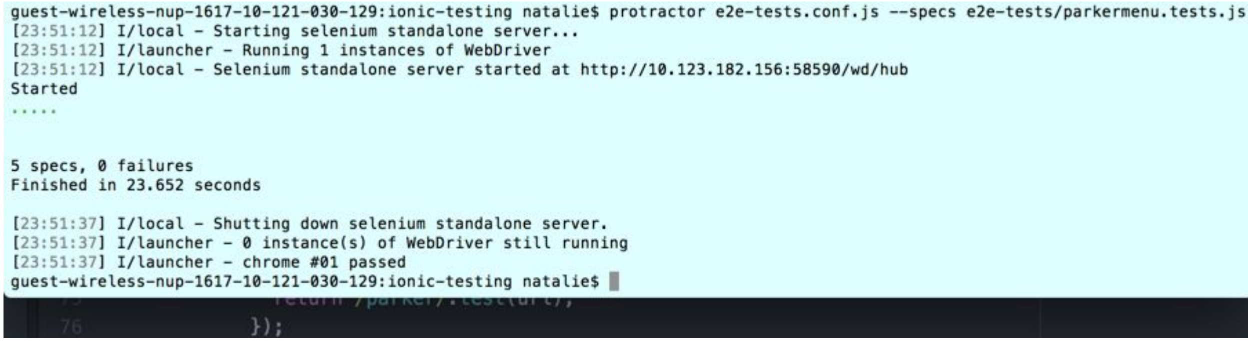
```
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$ protractor e2e-tests.conf.js --specs e2e-tests/parkermenu.tests.js
[23:51:12] I/local - Starting selenium standalone server...
[23:51:12] I/launcher - Running 1 instances of WebDriver
[23:51:12] I/local - Selenium standalone server started at http://10.123.182.156:58590/wd/hub
Started
.....


5 specs, 0 failures
Finished in 23.652 seconds

[23:51:37] I/local - Shutting down selenium standalone server.
[23:51:37] I/launcher - 0 instance(s) of WebDriver still running
[23:51:37] I/launcher - chrome #01 passed
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$
```

```
            });
```

    - All parkerMenu tests pass

        -

Test 3: Parker Menu option Search leads to the search page

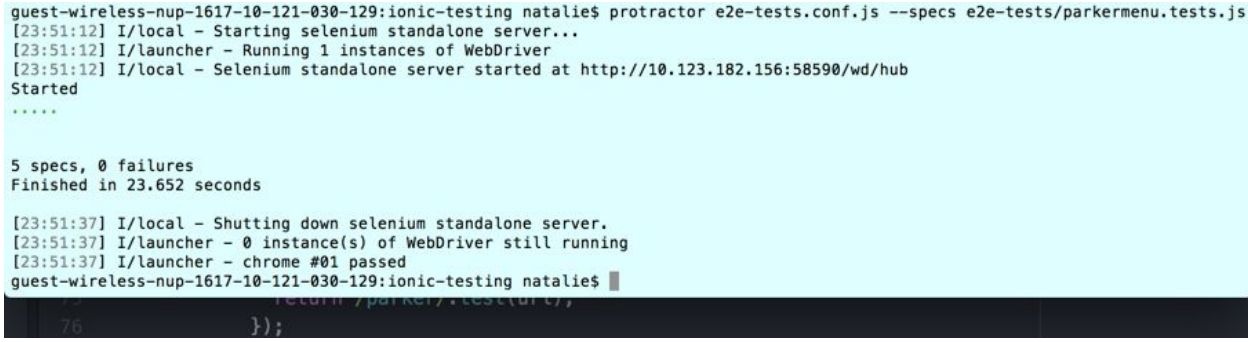
- Location: ParkHere/myApp/ionic-testing/e2e-tests/parkerMenu.tests.js
    - under it function 'parker menu Search --> search page'
- Description: When a signed in parker opens the app and they toggle open the menu, clicking on the Search option should lead them to the page that correctly displays search dialogue for the user to find a parking spot.
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/parkerMenu.tests.js
- Rationale: I correctly logged in under a parker account that I knew would work properly. From there I opened the menu and selecting map view and waited for the page to load (giving it a 5 second window to ensure that it was not too slow although every time it has been run so far it's been almost instantaneous).
- Results:
    - Since this test requires much automation, it’s not viewable via screenshot.
    - 

```
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$ protractor e2e-tests.conf.js --specs e2e-tests/parkermenu.tests.js
[23:51:12] I/local - Starting selenium standalone server...
[23:51:12] I/launcher - Running 1 instances of WebDriver
[23:51:12] I/local - Selenium standalone server started at http://10.123.182.156:58590/wd/hub
Started
.....


5 specs, 0 failures
Finished in 23.652 seconds

[23:51:37] I/local - Shutting down selenium standalone server.
[23:51:37] I/launcher - 0 instance(s) of WebDriver still running
[23:51:37] I/launcher - chrome #01 passed
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$
```

        - All parkerMenu tests pass

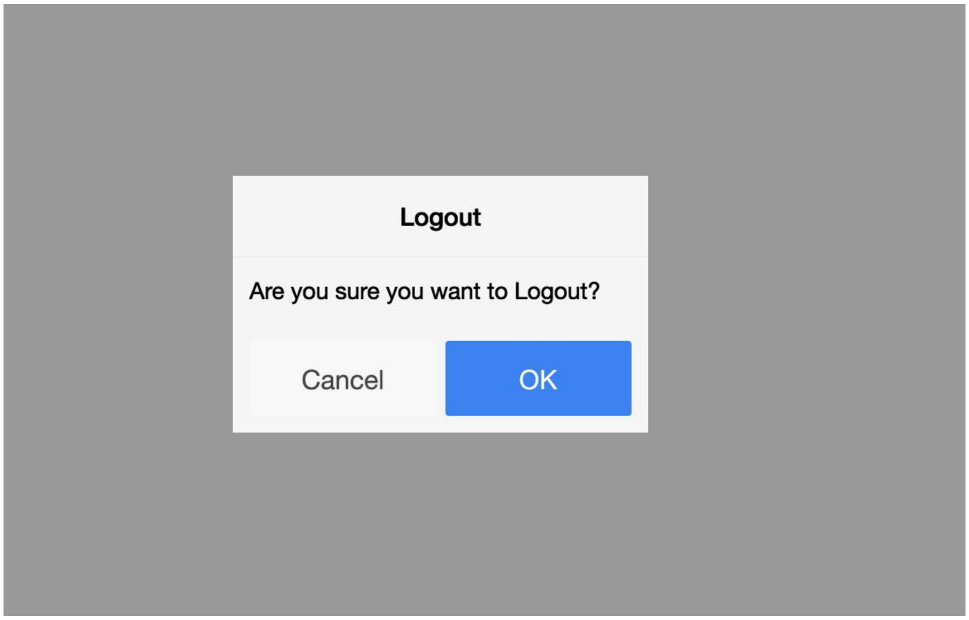
Test 4: Parker Menu option Logout correctly logs a parker out

- Location: ParkHere/myApp/ionic-testing/e2e-tests/parkerMenu.tests.js
    - under it function 'should log out of parker'
- Description: When a signed in parker opens the app and they toggle open the menu, clicking on the Logout option and selecting confirm at the popup dialogue should lead the parker to the login screen.
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/parkerMenu.tests.js
- Rationale: I correctly logged in under a parker account that I knew would work properly. From there I opened the menu and selected log out, pressed confirm when it verified, and was sent to the login page. (Note: You can only access the login page if a user is logged out (or else it directs you to the home page) so this indicates that it logs out the user).
- Results:

    - Since this test requires much automation, it’s not viewable via screenshot.
    - 

```
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$ protractor e2e-tests.conf.js --specs e2e-tests/parkermenu.tests.js
[23:51:12] I/local - Starting selenium standalone server...
[23:51:12] I/launcher - Running 1 instances of WebDriver
[23:51:12] I/local - Selenium standalone server started at http://10.123.182.156:58590/wd/hub
Started
.....


5 specs, 0 failures
Finished in 23.652 seconds

[23:51:37] I/local - Shutting down selenium standalone server.
[23:51:37] I/launcher - 0 instance(s) of WebDriver still running
[23:51:37] I/launcher - chrome #01 passed
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$
```

```
        });
```

        - All parkerMenu tests pass
    - 

Logout

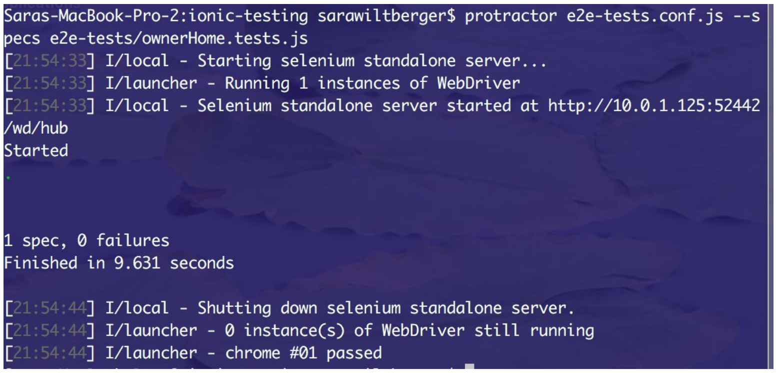
Are you sure you want to Logout?

        - Front-end of logout dialogue box

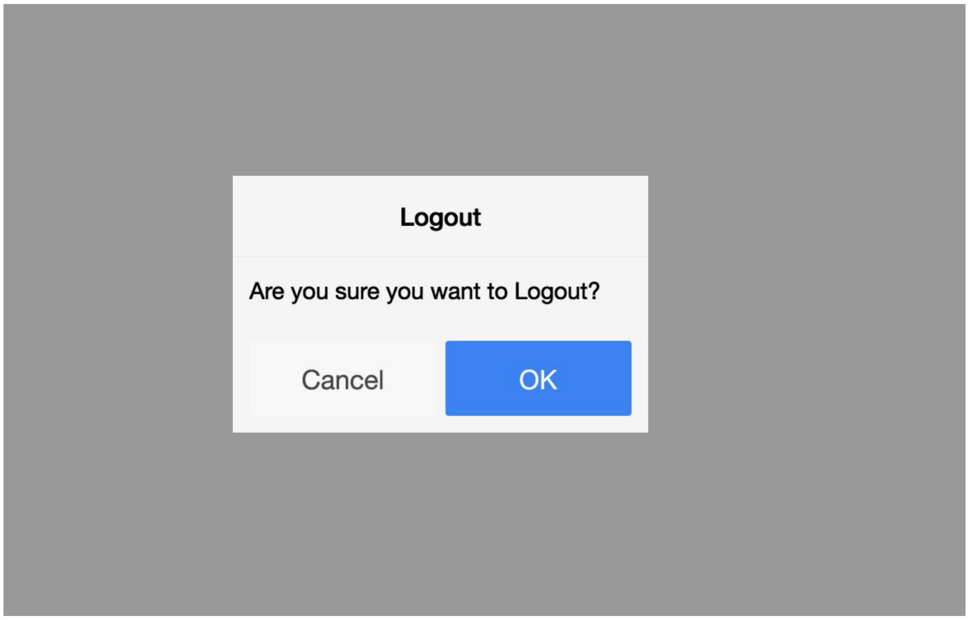
Test 5: Parker Menu option Logout does not log out a parker when canceled

- Location: ParkHere/myApp/ionic-testing/e2e-tests/parkerMenu.tests.js
    - under it function 'should not log out of parker if cancelled'
- Description: When a signed in parker opens the app and they toggle open the menu, clicking on the Logout option and selecting cancel at the popup dialogue should allow the parker to remain on the same screen.
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/parkerMenu.tests.js
- Rationale: I correctly logged in under a parker account that I knew would work properly. From there I opened the menu and selected log out, pressed cancel when it checked to verify, and remained on the same page. (Note: You can only access an internal page if a

user is logged in (or else it directs you to the login page) so this indicates that it does not logout the user).

- Results:
- 

```
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --s
pecs e2e-tests/ownerHome.tests.js
[21:54:33] I/local - Starting selenium standalone server...
[21:54:33] I/launcher - Running 1 instances of WebDriver
[21:54:33] I/local - Selenium standalone server started at http://10.0.1.125:52442
/wd/hub
Started
.


1 spec, 0 failures
Finished in 9.631 seconds

[21:54:44] I/local - Shutting down selenium standalone server.
[21:54:44] I/launcher - 0 instance(s) of WebDriver still running
[21:54:44] I/launcher - chrome #01 passed
```

  - Since this test requires much automation, it’s not viewable via screenshot.
  - All tests pass
  - 

    Logout

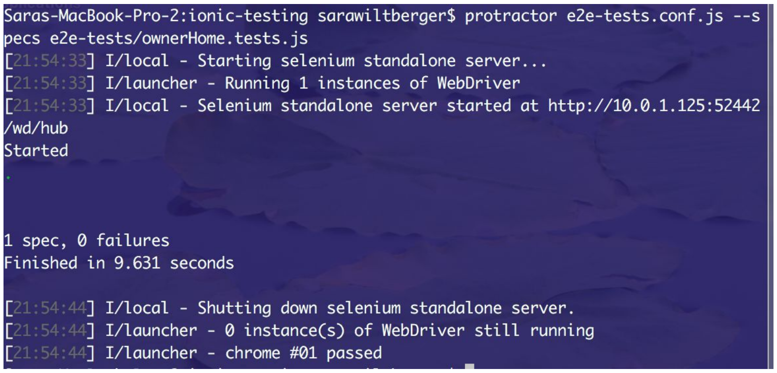
    Are you sure you want to Logout?

    - Front-end of logout dialogue box

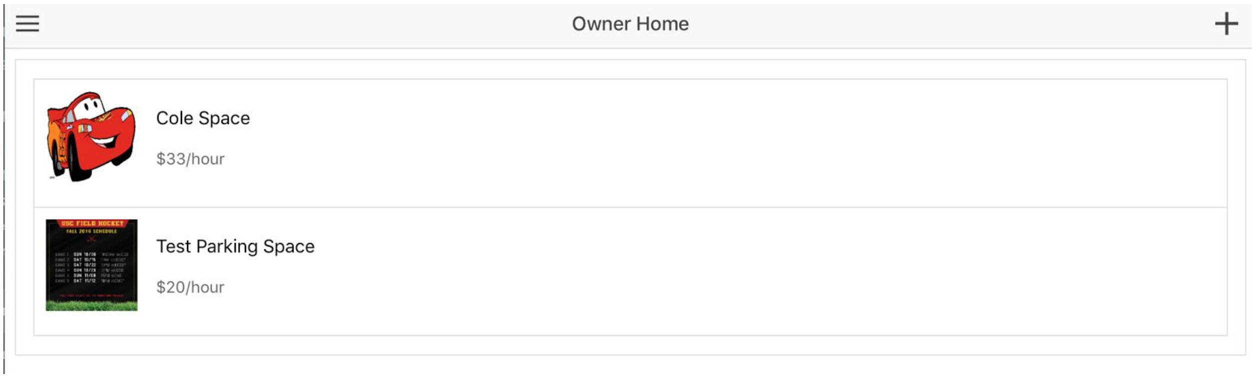
Owner Home Test

Test 1: Owner’s Reserved Spaces Show up

- Location: ParkHere/myApp/ionic-testing/e2e-tests/ownerHome.tests.js
  - under it function 'should automatically show owner spaces already populated'
- Description: When a signed in owner opens the app, they should be redirected to their homescreen and they should see a list of parking spaces they have uploaded
- Execution:
  - from ParkHere/myApp/ionic-testing
  - run: protractor e2e-tests.conf.js --specs e2e-tests/ownerHome.tests.js

- Rationale: I originally inputted two parking spaces one named “Cole World” at $33 an hour and one names “Test Parking Space” at $20 an hour. I wanted to make sure they both appeared when I logged in as this owner. I chose two spaces because one is too trivial and I wanted to make sure it wouldn’t break with multiple spaces
- Results:
- 

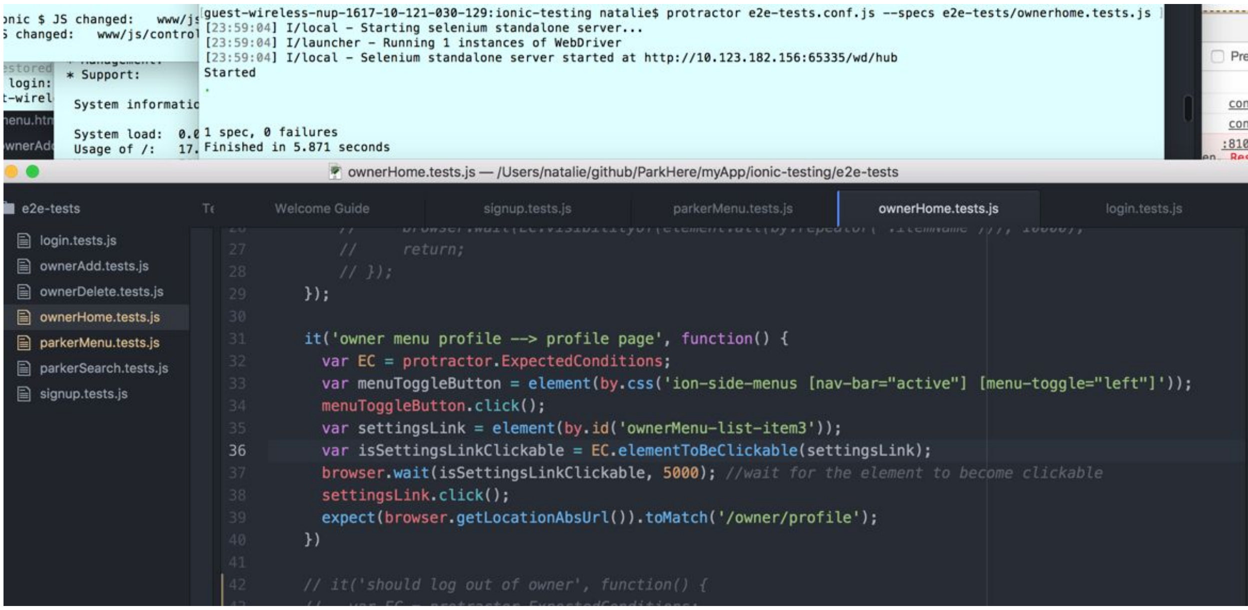
```
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --s
pecs e2e-tests/ownerHome.tests.js
[21:54:33] I/local - Starting selenium standalone server...
[21:54:33] I/launcher - Running 1 instances of WebDriver
[21:54:33] I/local - Selenium standalone server started at http://10.0.1.125:52442
/wd/hub
Started
.


1 spec, 0 failures
Finished in 9.631 seconds

[21:54:44] I/local - Shutting down selenium standalone server.
[21:54:44] I/launcher - 0 instance(s) of WebDriver still running
[21:54:44] I/launcher - chrome #01 passed
```

- 

  - The test passed with no failures. Both parking spaces appeared when I signed in

Test 2: Owner Menu option Profile correctly redirects to the owner’s Profile page

- Location: ParkHere/myApp/ionic-testing/e2e-tests/ownerMenu.tests.js
  - under it function 'owner menu profile --> profile page'
- Description: When a signed in owner opens the app and they toggle open the menu, clicking on the Profile option and redirects them to their profile page.
- Execution:
  - from ParkHere/myApp/ionic-testing
  - run: protractor e2e-tests.conf.js --specs e2e-tests/ownerHome.tests.js
- Rationale: I correctly logged in under an owner account that I knew would work properly. From there I opened the menu and selected profile and waited for the page to load (giving it a 5 second window to ensure that it was not too slow although every time it has been run so far it's been almost instantaneous).
- Results:
  - Since this test requires much automation, it’s not viewable via screenshot.

- 

```
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$ protractor e2e-tests.conf.js --specs e2e-tests/ownerhome.tests.js
[23:59:04] I/local - Starting selenium standalone server...
[23:59:04] I/launcher - Running 1 instances of WebDriver
[23:59:04] I/local - Selenium standalone server started at http://10.123.182.156:65335/wd/hub
Started
.


1 spec, 0 failures
Finished in 5.871 seconds
```

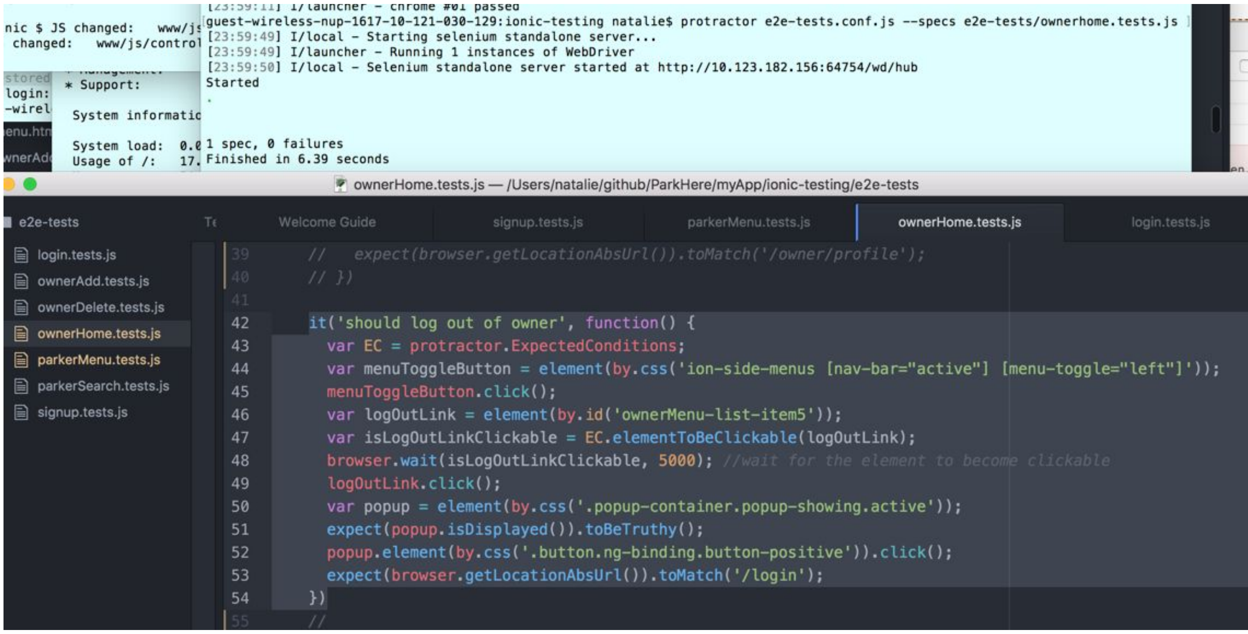
ownerHome.tests.js — /Users/natalie/github/ParkHere/myApp/ionic-testing/e2e-tests

```
    //     browser.wait(EC.visibilityOf(element.all(by.repeater('.itemName'))), 10000);
    //     return;
    // });
  });

  it('owner menu profile --> profile page', function() {
    var EC = protractor.ExpectedConditions;
    var menuToggleButton = element(by.css('ion-side-menus [nav-bar="active"] [menu-toggle="left"]'));
    menuToggleButton.click();
    var settingsLink = element(by.id('ownerMenu-list-item3'));
    var isSettingsLinkClickable = EC.elementToBeClickable(settingsLink);
    browser.wait(isSettingsLinkClickable, 5000); //wait for the element to become clickable
    settingsLink.click();
    expect(browser.getLocationAbsUrl()).toMatch('/owner/profile');
  })

  // it('should log out of owner', function() {
  //   var EC = protractor.ExpectedConditions;
```

    - ■ Test passes

Test 3: Owner Menu option Logout correctly logs an owner out

- Location: ParkHere/myApp/ionic-testing/e2e-tests/ownerHome.tests.js
    - under it function 'should log out of owner’
- Description: When a signed in owner opens the app and they toggle open the menu, clicking on the Logout option and selecting confirm at the popup dialogue should lead the owner to the login screen.
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/ownerHome.tests.js
- Rationale: I correctly logged in under an owner account that I knew would work properly. From there I opened the menu and selected log out, pressed confirm when it verified, and was sent to the login page. (Note: You can only access the login page if a user is logged out (or else it directs you to the home page) so this indicates that it logs out the user).
- Results:
    - Since this test requires much automation, it’s not viewable via screenshot.

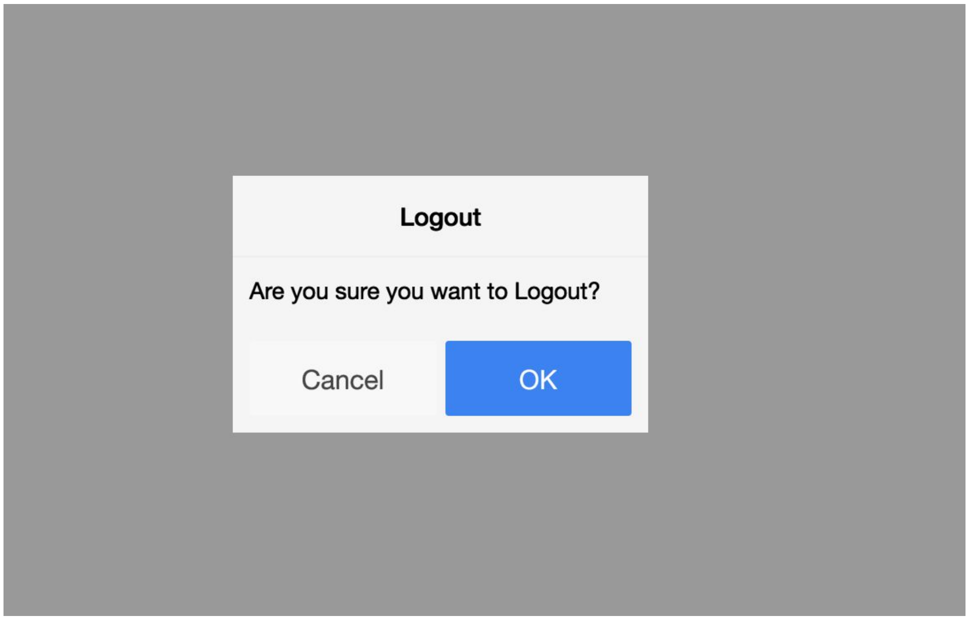
- 

```
[23:59:11] I/launcher - chrome #01 passed
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$ protractor e2e-tests.conf.js --specs e2e-tests/ownerhome.tests.js
[23:59:49] I/local - Starting selenium standalone server...
[23:59:49] I/launcher - Running 1 instances of WebDriver
[23:59:50] I/local - Selenium standalone server started at http://10.123.182.156:64754/wd/hub
Started
.


1 spec, 0 failures
Finished in 6.39 seconds
```

ownerHome.tests.js — /Users/natalie/github/ParkHere/myApp/ionic-testing/e2e-tests

```
    //   expect(browser.getLocationAbsUrl()).toMatch('/owner/profile');
    // })

    it('should log out of owner', function() {
      var EC = protractor.ExpectedConditions;
      var menuToggleButton = element(by.css('ion-side-menus [nav-bar="active"] [menu-toggle="left"]'));
      menuToggleButton.click();
      var logOutLink = element(by.id('ownerMenu-list-item5'));
      var isLogOutLinkClickable = EC.elementToBeClickable(logOutLink);
      browser.wait(isLogOutLinkClickable, 5000); //wait for the element to become clickable
      logOutLink.click();
      var popup = element(by.css('.popup-container.popup-showing.active'));
      expect(popup.isDisplayed()).toBeTruthy();
      popup.element(by.css('.button.ng-binding.button-positive')).click();
      expect(browser.getLocationAbsUrl()).toMatch('/login');
    })
    //
```

  - Test passes

- 

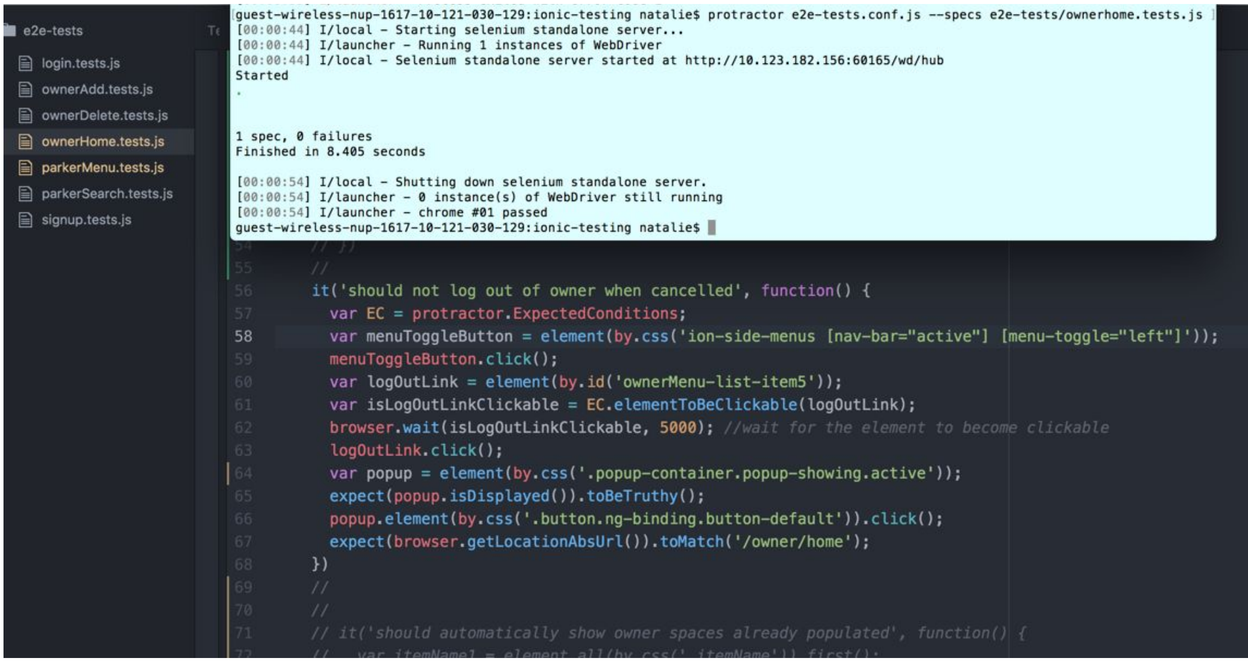
Logout

Are you sure you want to Logout?

  - Front-end of logout dialogue box

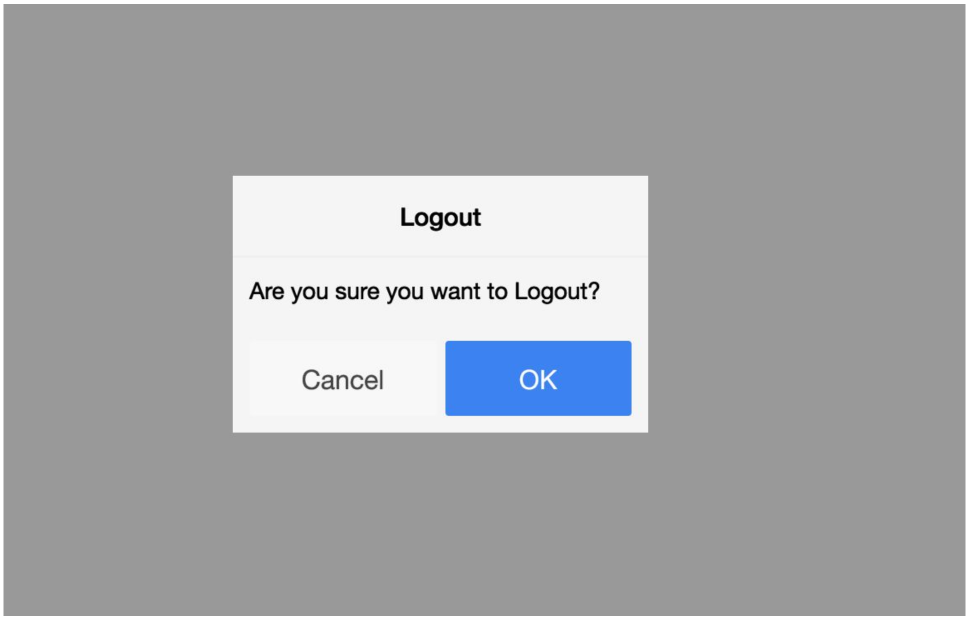
Test 4: Parker Menu option Logout does not log out an owner when cancelled

- Location: ParkHere/myApp/ionic-testing/e2e-tests/ownerHome.tests.js
  - under it function 'should not log out of owner when cancelled'
- Description: When a signed in owner opens the app and they toggle open the menu, clicking on the Logout option and selecting cancel at the popup dialogue should allow the parker to remain on the same screen.
- Execution:
  - from ParkHere/myApp/ionic-testing

    - run: protractor e2e-tests.conf.js --specs e2e-tests/ownerHome.tests.js
- Rationale: I correctly logged in under a owner account that I knew would work properly. From there I opened the menu and selected log out, pressed cancel when it checked to verify, and remained on the same page. (Note: You can only access an internal page if a user is logged in (or else it directs you to the login page) so this indicates that it does not logout the user).
- Results:
    - Since this test requires much automation, it's not viewable via screenshot.
    - 

```
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$ protractor e2e-tests.conf.js --specs e2e-tests/ownerhome.tests.js
[00:00:44] I/local - Starting selenium standalone server...
[00:00:44] I/launcher - Running 1 instances of WebDriver
[00:00:44] I/local - Selenium standalone server started at http://10.123.182.156:60165/wd/hub
Started
.


1 spec, 0 failures
Finished in 8.405 seconds

[00:00:54] I/local - Shutting down selenium standalone server.
[00:00:54] I/launcher - 0 instance(s) of WebDriver still running
[00:00:54] I/launcher - chrome #01 passed
guest-wireless-nup-1617-10-121-030-129:ionic-testing natalie$
```

```
    //
    it('should not log out of owner when cancelled', function() {
      var EC = protractor.ExpectedConditions;
      var menuToggleButton = element(by.css('ion-side-menus [nav-bar="active"] [menu-toggle="left"]'));
      menuToggleButton.click();
      var logOutLink = element(by.id('ownerMenu-list-item5'));
      var isLogOutLinkClickable = EC.elementToBeClickable(logOutLink);
      browser.wait(isLogOutLinkClickable, 5000); //wait for the element to become clickable
      logOutLink.click();
      var popup = element(by.css('.popup-container.popup-showing.active'));
      expect(popup.isDisplayed()).toBeTruthy();
      popup.element(by.css('.button.ng-binding.button-default')).click();
      expect(browser.getLocationAbsUrl()).toMatch('/owner/home');
    })
    //
    //
    // it('should automatically show owner spaces already populated', function() {
```

        - Test passes
    - 

Logout

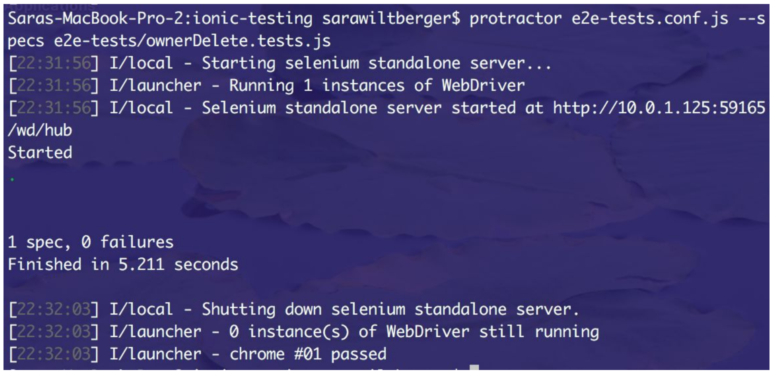
Are you sure you want to Logout?

        - Front-end of logout dialogue box

Owner Home Test Delete

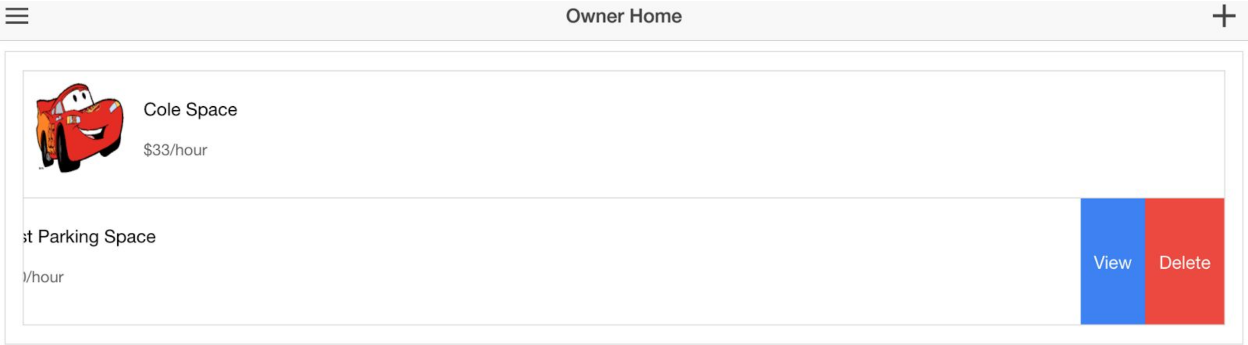
Test 1: Owner Delete A Spot with no reservations

- Location: ParkHere/myApp/ionic-testing/e2e-tests/ownerDelete.tests.js
    - under it function 'successful delete (no parkers)'
- Description: An owner should be able to delete an added parking space if no parkers have yet reserved it
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/ownerDelete.tests.js
- Rationale: I try to delete a space I know no one has reserved. This should pop up a confirmation confirming that I want to delete. My inputs are basically swiping, deleting, and confirming. In my code, I confirmed that the space was there, that a popup resulted when I pushed delete, and that there was only one space left after I deleted it.
- Results:

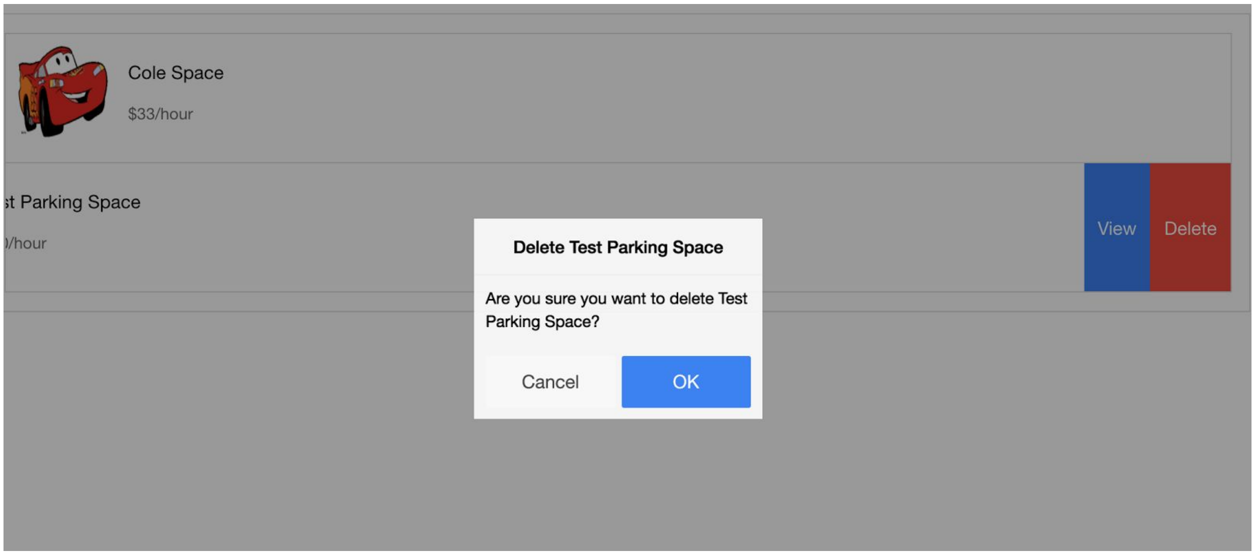


```
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --s
pecs e2e-tests/ownerDelete.tests.js
[22:31:56] I/local - Starting selenium standalone server...
[22:31:56] I/launcher - Running 1 instances of WebDriver
[22:31:56] I/local - Selenium standalone server started at http://10.0.1.125:59165
/wd/hub
Started
.


1 spec, 0 failures
Finished in 5.211 seconds

[22:32:03] I/local - Shutting down selenium standalone server.
[22:32:03] I/launcher - 0 instance(s) of WebDriver still running
[22:32:03] I/launcher - chrome #01 passed
```

-

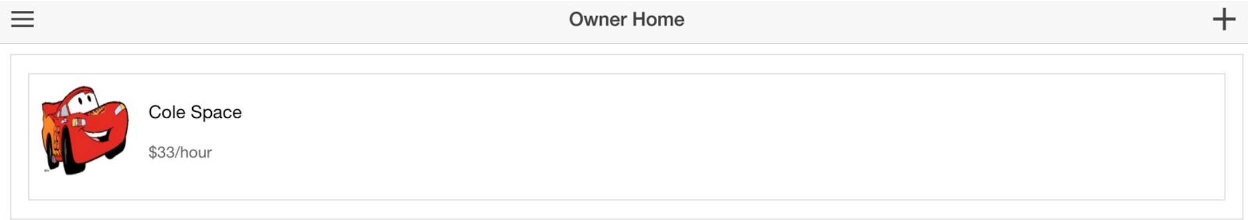


-

- 

  Delete Test Parking Space

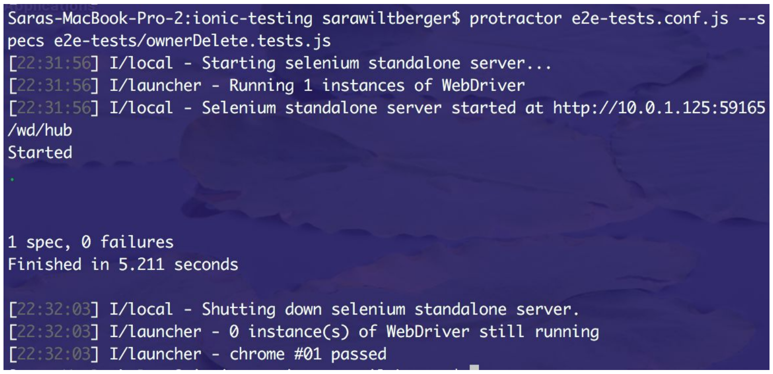
  Are you sure you want to delete Test Parking Space?

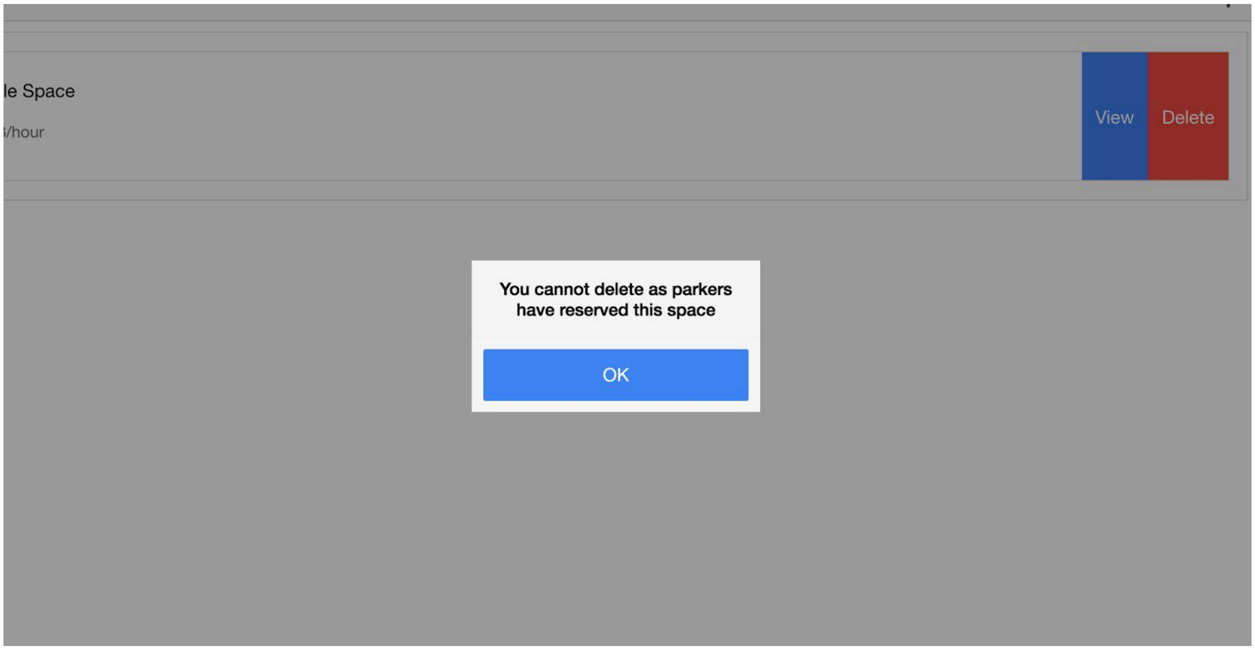
-
  - The test passed with no failures. I successfully deleted the space and it no longer showed up on the front end.

Test 2: Owner delete a spot with reservations

- Location: ParkHere/myApp/ionic-testing/e2e-tests/ownerDelete.tests.js
  - under it function 'cannot delete because of reservations'
- Description: An owner cannot delete a parking space if owners have reserved it. A warning popup will appear.
- Execution:
  - from ParkHere/myApp/ionic-testing
  - run: protractor e2e-tests.conf.js --specs e2e-tests/ownerDelete.tests.js
- Rationale: I try to delete a space I know is reserved. A popup will show telling the user that this is not allowed
- Results:

- 

```
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --s
pecs e2e-tests/ownerDelete.tests.js
[22:31:56] I/local - Starting selenium standalone server...
[22:31:56] I/launcher - Running 1 instances of WebDriver
[22:31:56] I/local - Selenium standalone server started at http://10.0.1.125:59165
/wd/hub
Started
.


1 spec, 0 failures
Finished in 5.211 seconds

[22:32:03] I/local - Shutting down selenium standalone server.
[22:32:03] I/launcher - 0 instance(s) of WebDriver still running
[22:32:03] I/launcher - chrome #01 passed
```

- 

  You cannot delete as parkers have reserved this space

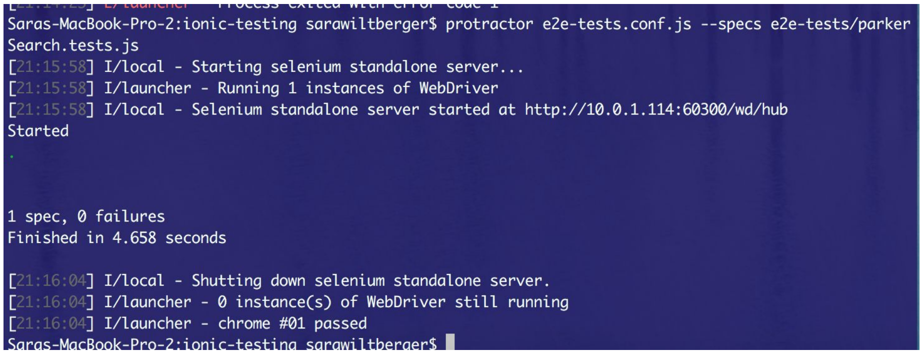
  - The test passed with no failures. I could not delete the space I knew had reservations

Parker Search

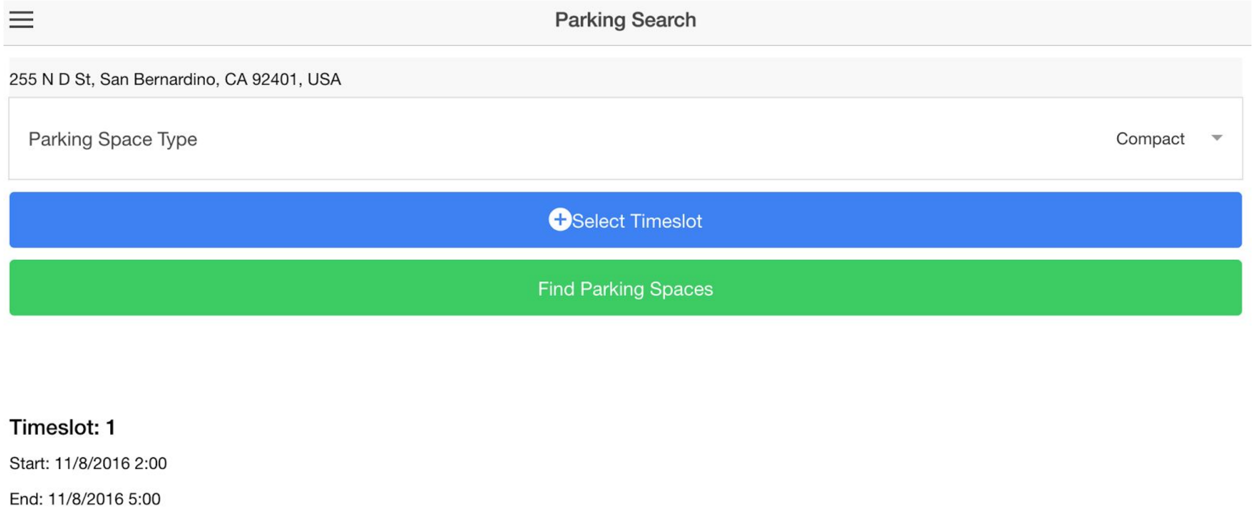
Test 1: Searching for a parking Space

- Location: ParkHere/myApp/ionic-testing/e2e-tests/parkerSearch.tests.js
  - under it function ‘search should produce the correct space'
- Description: Searching for a parking space I know exists in the database should appear on the search results screen
- Execution:
  - from ParkHere/myApp/ionic-testing
  - run: protractor e2e-tests.conf.js --specs e2e-tests/parkerSearch.tests.js
    - Note: these will run all the parkerSearch tests

    - because the testing framework we choose does not work with the ionic popup plug in, I have to pause the test and let the user manually input the space start and end time and date.
- Rationale: I inputed “255 N D St, San Bernardino, CA 92401, USA” as the address, chose ‘compact’ as the type, Nov 8th 2 o’clock as the start, and Nov 8th 3 o’clock as the end. I used this input because I knew it was a space already saved in the database, and I wanted to confirm that a parker could find it
- Results:



```
[21:14:25] E/launcher - Process exited with error code 1
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --specs e2e-tests/parker
Search.tests.js
[21:15:58] I/local - Starting selenium standalone server...
[21:15:58] I/launcher - Running 1 instances of WebDriver
[21:15:58] I/local - Selenium standalone server started at http://10.0.1.114:60300/wd/hub
Started
.


1 spec, 0 failures
Finished in 4.658 seconds

[21:16:04] I/local - Shutting down selenium standalone server.
[21:16:04] I/launcher - 0 instance(s) of WebDriver still running
[21:16:04] I/launcher - chrome #01 passed
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$
```



Parking Search

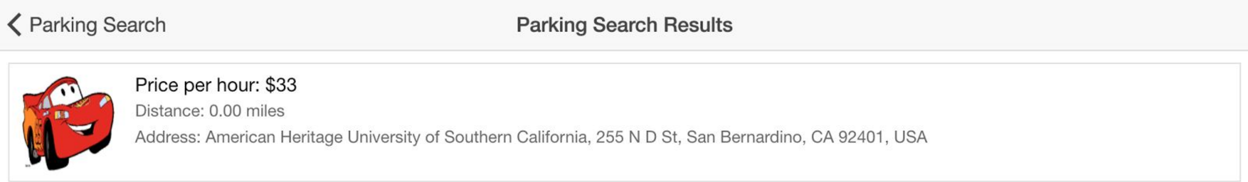
255 N D St, San Bernardino, CA 92401, USA

Parking Space Type: Compact

Timeslot: 1

Start: 11/8/2016 2:00

End: 11/8/2016 5:00



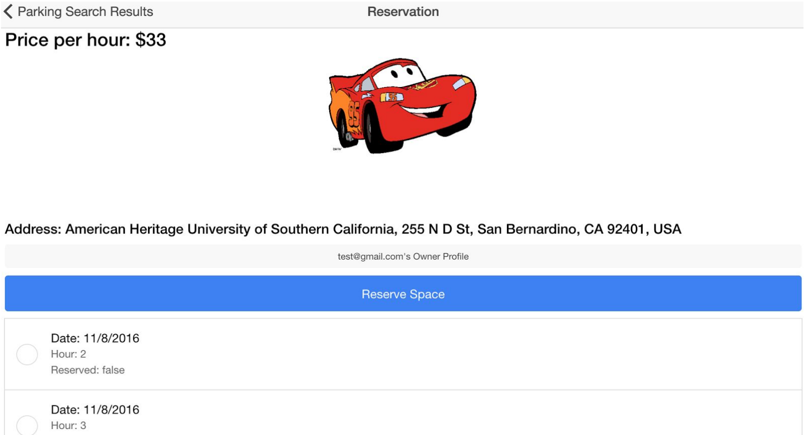
Parking Search

Parking Search Results

Price per hour: $33

Distance: 0.00 miles

Address: American Heritage University of Southern California, 255 N D St, San Bernardino, CA 92401, USA

- 

  Parking Search Results

  Reservation

  Price per hour: $33

  Address: American Heritage University of Southern California, 255 N D St, San Bernardino, CA 92401, USA

  test@gmail.com's Owner Profile

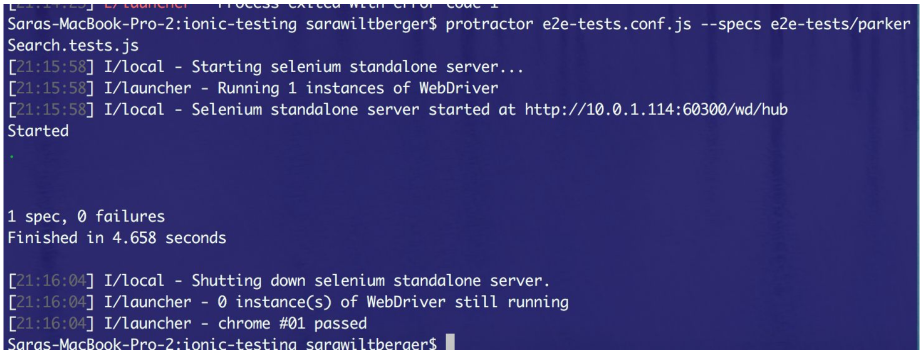
  Reserve Space

  ○ Date: 11/8/2016
  Hour: 2
  Reserved: false

  ○ Date: 11/8/2016
  Hour: 3
-
  - Searching for the parking space produced the correct result

Test 2: Searching for a parking Space within 3 miles

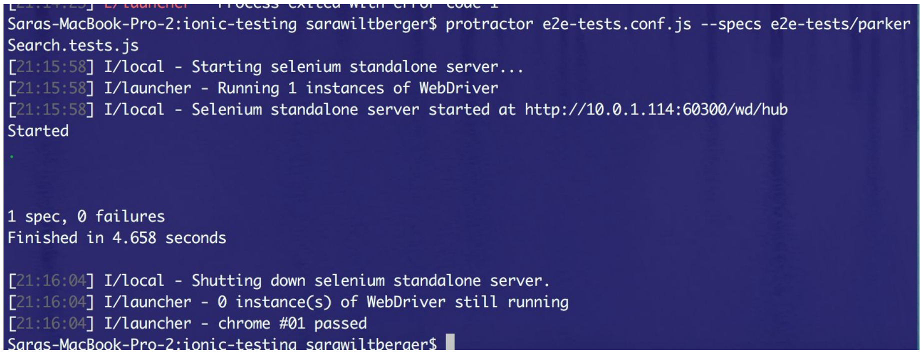
- Location: ParkHere/myApp/ionic-testing/e2e-tests/parkerSearch.tests.js
  - under it function ‘search should search within 3 miles'
- Description: Same as above search, instead I look for a parking space that is within a 3 mile radius of the address of the parking space
- Execution:
  - from ParkHere/myApp/ionic-testing
  - run: protractor e2e-tests.conf.js --specs e2e-tests/parkerSearch.tests.js
    - Note: these will run all the parkerSearch tests
  - because the testing framework we choose does not work with the ionic popup plug in, I have to pause the test and let the user manually input the space start and end time and date.
- Rationale: The inputs were exactly the same as above, except the address inputted was ”69 E Central Ave, San Bernardino, CA 92408” as this was 1.7 miles away from the address of the parking space I was looking for
- Results:

- 

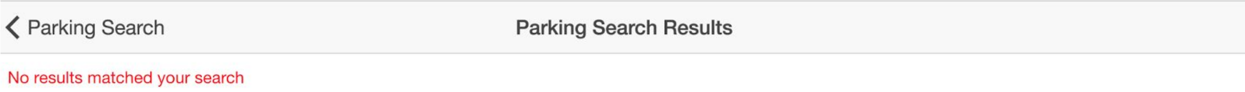
```
[21:14:23] E/launcher - Process exited with error code 1
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --specs e2e-tests/parker
Search.tests.js
[21:15:58] I/local - Starting selenium standalone server...
[21:15:58] I/launcher - Running 1 instances of WebDriver
[21:15:58] I/local - Selenium standalone server started at http://10.0.1.114:60300/wd/hub
Started
.


1 spec, 0 failures
Finished in 4.658 seconds

[21:16:04] I/local - Shutting down selenium standalone server.
[21:16:04] I/launcher - 0 instance(s) of WebDriver still running
[21:16:04] I/launcher - chrome #01 passed
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$
```

- The results were exactly the same as above which indicates a correct output

Test 3: Search produces no results:

- Location: ParkHere/myApp/ionic-testing/e2e-tests/parkerSearch.tests.js
    - under it function ‘search should search within 3 miles'
- Description: If the parking space search has no valid matches, there should be an indicator of this on the results page
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/parkerSearch.tests.js
        - Note: these will run all the parkerSearch tests
- Rationale: The inputs were exactly the same as above, except the address inputted was ”111 North Pole” which I knew was not within 3 miles of any space in the database
- Results:
    - 

```
[21:14:23] E/launcher - Process exited with error code 1
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ protractor e2e-tests.conf.js --specs e2e-tests/parker
Search.tests.js
[21:15:58] I/local - Starting selenium standalone server...
[21:15:58] I/launcher - Running 1 instances of WebDriver
[21:15:58] I/local - Selenium standalone server started at http://10.0.1.114:60300/wd/hub
Started
.


1 spec, 0 failures
Finished in 4.658 seconds

[21:16:04] I/local - Shutting down selenium standalone server.
[21:16:04] I/launcher - 0 instance(s) of WebDriver still running
[21:16:04] I/launcher - chrome #01 passed
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$
```



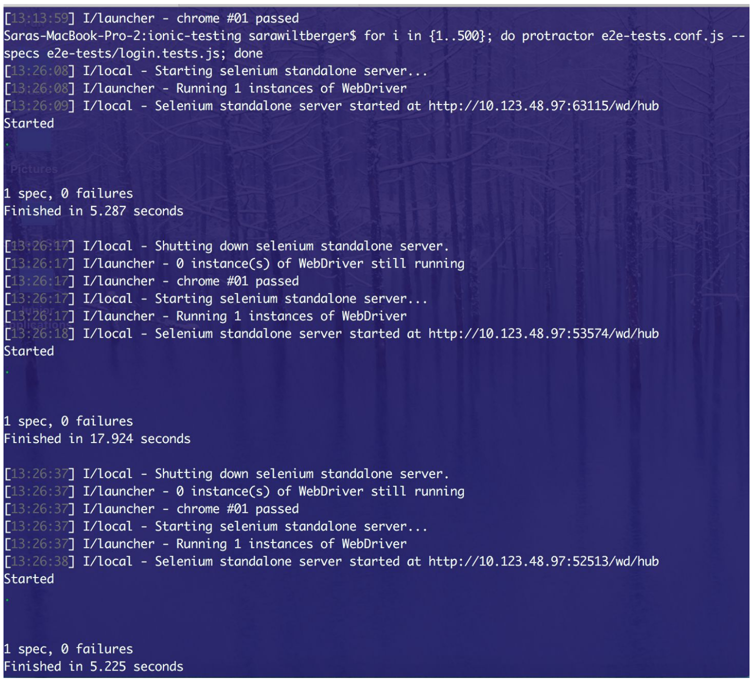
No results matched your search

-

    - A message indicating that no results return appeared

Non-functional Requirements:

Test 1: Scale:

- Location: ParkHere/myApp/ionic-testing/e2e-tests/login.tests.js
    - under it function 'successful delete (no parkers)'
- Description: Our requirements specified that the app should be "scalable." I tested logging in 500 users by running the below script. Since I did not log out, it created 500 sessions and allowed 500 users to be logged in at once
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: for i in {1..500}; do protractor e2e-tests.conf.js --specs e2e-tests/login.tests.js; done
- Rationale: I chose 500 users as this was defined in our requirements doc Also, the test took a while and I wanted to use my computer for other things, so I decided an hour was the limit of testing for this section. Each test took about 5 secs each to run in the framework so I only tested logging in and no other functionality. I decided to test logging in to ensure that the server could handle these 500 users
- Results:
    - 

```
[13:13:59] I/launcher - chrome #01 passed
Saras-MacBook-Pro-2:ionic-testing sarawiltberger$ for i in {1..500}; do protractor e2e-tests.conf.js --
specs e2e-tests/login.tests.js; done
[13:26:08] I/local - Starting selenium standalone server...
[13:26:08] I/launcher - Running 1 instances of WebDriver
[13:26:09] I/local - Selenium standalone server started at http://10.123.48.97:63115/wd/hub
Started
.


1 spec, 0 failures
Finished in 5.287 seconds

[13:26:17] I/local - Shutting down selenium standalone server.
[13:26:17] I/launcher - 0 instance(s) of WebDriver still running
[13:26:17] I/launcher - chrome #01 passed
[13:26:17] I/local - Starting selenium standalone server...
[13:26:17] I/launcher - Running 1 instances of WebDriver
[13:26:18] I/local - Selenium standalone server started at http://10.123.48.97:53574/wd/hub
Started
.


1 spec, 0 failures
Finished in 17.924 seconds

[13:26:37] I/local - Shutting down selenium standalone server.
[13:26:37] I/launcher - 0 instance(s) of WebDriver still running
[13:26:37] I/launcher - chrome #01 passed
[13:26:37] I/local - Starting selenium standalone server...
[13:26:37] I/launcher - Running 1 instances of WebDriver
[13:26:38] I/local - Selenium standalone server started at http://10.123.48.97:52513/wd/hub
Started
.


1 spec, 0 failures
Finished in 5.225 seconds
```

    - These were the first three results out of the 500. All of the other tests were very similar.

Test 2: Fast: load within 2 seconds

- Location: ParkHere/myApp/ionic-testing/e2e-tests/parkerSearch.tests.js
    - under it function 'successful delete (no parkers)'
- Description: Our requirements specified that the app should never have a load time less than 2 sec. I could not think of a great way to test this, so I decided to try to time the load time of the section that took the longest to query
- Execution:
    - from ParkHere/myApp/ionic-testing
    - run: protractor e2e-tests.conf.js --specs e2e-tests/parkerSearch.tests.js
        - Note: these will run all the parkerSearch tests; I commented out the other two tests, to run this particular one
    - because the testing framework we choose does not work with the ionic popup plug in, I have to pause the test and let the user manually input the space start and end time and date.
    - After inputting the popup info, I ran a timer and timed how long it took the program to populate the search results
- Rationale: I chose to test the parker search, as this would take the most amount of time to query the data looked for compared to all the other functionality in the app
- Results:
    - I did the process three times, and the time it took for the program to display results from the time the button was pressed was .5, .7, and .8 seconds, all under 2 seconds.

Non-functional requirements we tested without a framework

- User-friendly:
    - I asked my roommate to try the app (with no explanations) and I timed her. She was able to figure out most of the features of parker side (we did not get to rating) within 5 minutes.

Non-function requirements we could not test:

- Efficient in Energy:
    - The only frameworks I could find that tested energy used had to be run on an actual Android device. No one in Team 3 owned an Android, so we could not test this

Whitebox Tests

Execution for **all whitebox tests**:

- from ParkHere/myApp/tests
- run: karma start unit-tests.conf.js

Login Controller

Test 1: Check if the scope of the login controller initializes to null

- Location: ParkHere/myApp/tests/unit-tests/login.controller.tests.js
- Description: We check if the scope variable has value of null in the instance of the login controller that is passed after initialization.
- Rationale: This checks that the login controller correctly initializes, before we have defined any scope variables. If it were to return false, we would know that our login controller was the source of a break in our code.

Test 2: Setting the username updates the user variable correctly

- Location: ParkHere/myApp/tests/unit-tests/login.controller.tests.js
- Description: When we call the setUsername function and pass a string, it should update the username string in user to equal that value.
- Rationale: This test evaluates that the setUsername function operates as intended, which is an essential function for updating the current user instance at login.

UpcomingSpaces Controller

Test 1: Check if owner can delete a spot that they've posted (making it unavailable) by swiping the spot on their profile listings

- Location: ParkHere/myApp/tests/unit-tests/upcomingspaces.controller.tests.js
- Description: We check the value of the scope variable "listCanSwipe," which is a boolean value that must be set to true when UpcomingSpacesCtrl is first entered in order for the swipe logic to execute.
- Rationale: This variable should be set to true at initialization in order to accomplish our app's functionality of allowing an owner to delete a spot posting on swipe. The logic for an owner to be able to do this is only executed if listCanSwipe is set to true in this case. We will want to test this whenever we add new code to ensure we are not breaking this functionality.

Map Controller

Test 1: Timeout option in Map controller

- Location: ParkHere/myApp/tests/unit-tests/map.controller.tests.js
- Description: Check if the timeout variable is set to 10000.
- Rationale: If the map search passes the timeout threshold then we want to stop the initialization of the map, inferring that the process is not running as expected. Our timeout variable ensures this threshold is accurately set, so we want to check that it is correct and unchanged whenever we push code.

Test 2: Check that high accuracy setting is enabled on maps at initialization

- Location: ParkHere/myApp/tests/unit-tests/map.controller.tests.js

- Description: We check that the boolean variable enableHighAccuracy is equal to true at the instance of controller that exists after initialization.
- Rationale: We want to verify that our map defaults to the high accuracy setting, only disabling when we search to another address. This intended behavior optimizes the map rendering in our map view.

ParkerSearch Controller

Test 1: List of search results should initialize to empty list

- Location: ParkHere/myApp/tests/unit-tests/parkersearch.controller.tests.js
- Description: The length of the list of search result should be zero at initialization.
- Rationale: This test simulates the functions that add to the list of search results in our controller, which ultimately is the basis for populating the search results page when a parker searches for a spot. We need to test that our service properly stores this information because it is the essential structure and logic behind our search results page.

Test 2: Add parking space to list of search results/store parking spaces accurately in list of search results

- Location: ParkHere/myApp/tests/unit-tests/parkersearch.controller.tests.js
- Description: The length of the list of search result should increment when a function pushes a parking space to the list.
- Rationale: This test simulates the functions that add to the list of search results in our controller, which ultimately is the basis for populating the search results page when a parker searches for a spot. We need to test that our service properly stores this information because it is the essential structure and logic behind our search results page.

Test 3: Remove parking space to list of search results

- Location: ParkHere/myApp/tests/unit-tests/parkersearch.controller.tests.js
- Description: The length of the list of search result should decrement when a function removes a parking space to the list.
- Rationale: This test simulates the removal of a parkingSpace from the list of search results in our controller, which ultimately is the basis for populating the search results page when a parker searches for a spot (as stated above). We need to test that our service properly stores this information for the same reason as described in test 2.

Test 4: Check that date (to be used in date selection tool) correctly initializes at the current date

- Location: ParkHere/myApp/tests/unit-tests/parkersearch.controller.tests.js
- Description: Our startDate (at initialization) must equal the accurate current date (which is the date that will be returned at call of newDate() function).

- Rationale: The date selection tool should automatically begin at the current date (at which point the user can update it with their date input), so we want to test that the initialization occurs correctly.

Test 5: Verify that geopoint variable is of type GeoPoint() upon initialization

- Location: ParkHere/myApp/tests/unit-tests/parkersearch.controller.tests.js
- Description: We check that the geopoint parkerSearch service variable is an instance of Parse.GeoPoint().
- Rationale: We need the variable to be of the correct type in order to perform operations on it correctly in our controller. We can test this at initialization, once the type is specified as we have created an instance of the variable.

Reservation Controller

Test 1: Check that price updates properly with owner input

- Location: ParkHere/myApp/tests/unit-tests/reservation.controller.tests.js
- Description: When we call the setPrice function we want it to correctly update the reservationInfo's price to equal the price we chose to set it to.
- Rationale: This test evaluates whether our setPrice function correctly allows us to change the price of a spot (and update reservationInfo accordingly). Not only should the price change upon call of this function, but it should change to equal the desired spot.

Test 2: Check that name of a space updates properly with owner input

- Location: ParkHere/myApp/tests/unit-tests/reservation.controller.tests.js
- Description: Calling the setSpotName function should to correctly update the reservationInfo's name string to equal the string we chose to set it to.
- Rationale: This test evaluates the correctness of our setSpotName function, checking that it performs its intended behavior of updating the name in the reservationInfo, which is essential functionality for the owner to be able to update relevant information about their parking spot.

ParkingSpace Controller

Test 1: Verify that parkingSpace is empty on initialization

- Location: ParkHere/myApp/tests/unit-tests/parkingspace.controller.tests.js
- Description: We check that a newly initialized parkingSpace is equal to null.
- Rationale: We want to make sure that the parkingSpace object that we created correctly initializes to null so that we can update the object without having to deal with junk values or other incorrect initialization errors.

Test 2: Verify that setEmail correctly updates parkingSpace email string

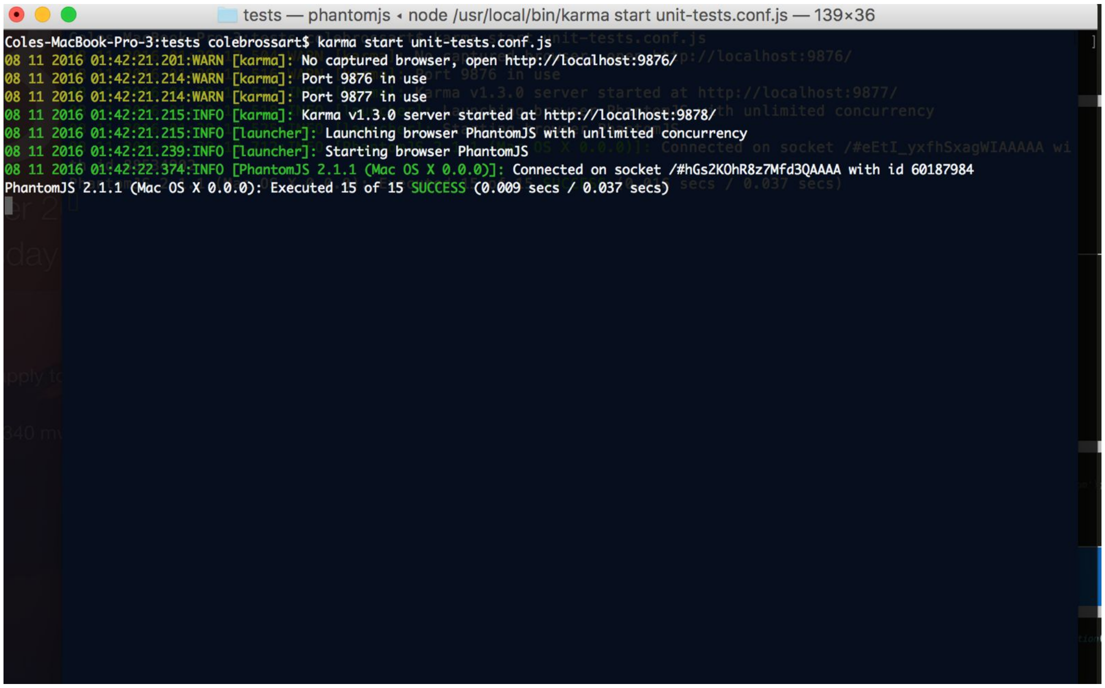
- Location: ParkHere/myApp/tests/unit-tests/parkingspace.controller.tests.js
- Description: We check that, after a call to setEmail with a string passed, the email string in a parkingSpace is equal to the value that we passed.
- Rationale: We need to verify that our setEmail function works as intended, meaning that it correctly updates the email information in a parkingSpace. This function is necessary for an owner to update the information associated with their parking spot, so we must test that it works as intended in order to verify that an owner account has all of our intended functionality.

OwnerHome Controller

Test 2: Test that setURL correctly sets the URL value

- Location: ParkHere/myApp/tests/unit-tests/ownerhome.controller.tests.js
- Description: We test that after calling setURL and passing a string value, the parkingSpace URL value is equal to the string that we passed.
- Rationale: We must verify that the setURL function correctly updates the value stored in parkingSpace URL string. This function is essential for storing information relevant to the parkingSpace object.

Results for **all whitebox tests**:


tests — phantomjs ◂ node /usr/local/bin/karma start unit-tests.conf.js — 139×36

```
Coles-MacBook-Pro-3:tests colebrossart$ karma start unit-tests.conf.js
08 11 2016 01:42:21.201:WARN [karma]: No captured browser, open http://localhost:9876/
08 11 2016 01:42:21.214:WARN [karma]: Port 9876 in use
08 11 2016 01:42:21.214:WARN [karma]: Port 9877 in use
08 11 2016 01:42:21.215:INFO [karma]: Karma v1.3.0 server started at http://localhost:9878/
08 11 2016 01:42:21.215:INFO [launcher]: Launching browser PhantomJS with unlimited concurrency
08 11 2016 01:42:21.239:INFO [launcher]: Starting browser PhantomJS
08 11 2016 01:42:22.374:INFO [PhantomJS 2.1.1 (Mac OS X 0.0.0)]: Connected on socket /#hGs2KOhR8z7Mfd3QAAAA with id 60187984
PhantomJS 2.1.1 (Mac OS X 0.0.0): Executed 15 of 15 SUCCESS (0.009 secs / 0.037 secs)
```

All tests (15) pass as expected.