

Test Plan for GreyMatter's FlyAway

Use Case Testing (5-10)

Title: Create Account

Actors: User (a generalization of Pilot and Passenger) and Account Authentication

Main Scenario:

- 1: User enters email, password and type
- 2: The system asks if the email is open
- 3: The Account Authentication indicates that the email and username are available
- 4: The system sends the email, password and type to Account Authentication
- 5: Account Authentication creates an account with the specifications
- 6: Account Authentication gives the User access to the account
- 7: The User can access their new account

Alternatives:

- 3a: Account Authentication indicates the email is attached to an existing account
- 3a1: The User is asked to enter a different email
- 3b: Account Authentication indicates the email is improper
- 3b1: The User is asked to enter a proper email
- 5a: Account Authentication the password is improper
- 5a1: The account is not created and the User is asked to enter a proper password

Test Situations:

- 1) The email is not attached to an existing account and all inputs are proper
- 2) The email is attached to an existing account
- 3) The email is improper
- 4) The password is improper

Solution:

For every situation that is not the main scenario (2-4 test situations) the account is not created and a message is displayed to the User under the problematic text input box that explains what the error is and what should be done.

Test Coverage:

Base: number of main and alternative scenarios = 4

Test situations cover all 4 cases

100% coverage of use case

Title: Edit Passenger Account

Actors: Passenger

Main Scenario:

- 1: The Passenger clicks the edit button
- 2: the system displays the passenger edit page filled with what exists previously
- 3: The Passenger can edit what is in the input boxes
- 4: The Passenger clicks save
- 5: The system saves the text from the input boxes to the account

6: The system displays "changes saved"

Alternatives:

- 1a: The Passenger clicks the continue button from account setup
- 1a1: The continues the same as before from the different starting point
- 2a: The system displays empty boxes if nothing exists previously
- 4a: The Passenger clicks cancel
- 5a: The system discards the text from the input boxes
- 6a: The system displays "changes canceled"

Test Situations:

- 1) The Passenger starts from the edit button on their account
- 2) The Passenger starts from the continue button during account setup
- 3) The system has previous input to display
- 4) the system does not have previous input to display
- 5) The Passenger clicks the save button
- 6) The Passenger clicks the cancel button
- 7) The system saves the text from input boxes
- 8) The system discards the text from input boxes
- 9) The system displays "changes saved"
- 10) The system displays "changes canceled"

Solution:

For test situations 1 and 2, the two different buttons should start or call the beginning of the same process. For situations 3 and 4, if there is no previous input for a text box, the system treats it like the previous input is just an empty string. At 5 and 6, the system should branch to the method that gets called on click. For 7 and 9, they follow from 5. For 8 and 10, they follow from 6.

Test Coverage:

Base: number of main and alternative scenarios = 6

Test situations cover all 6 cases

100% coverage of use case

Title: Contact a Passenger

Actors: Pilot

Main Scenario:

- 1: The pilot clicks accept message request button from message request
- 2: The system sets up a message channel between the pilot and passenger
- 3: The system notifies the passenger and the pilot of the change
- 4: The pilot get s a success message and can now access the message channel to that passenger account

Alternatives:

- 1a: The pilot clicks reject message request button from message request
- 1a1: the system dismisses the message request and the following steps do not occur

Test Situations:

- 1) The pilot clicks accept on message request
- 2) The pilot clicks reject on message request

Solution:

For situation 2 the message request is dismissed by the system marking it as rejected and the pilot gets a rejection success message.

Test Coverage:

Base: number of main and alternative scenarios = 2

Test situations cover all 2 cases

100% coverage of use case

Title: Contact Pilot

Actors: Passenger and Pilot

Main Scenario:

- 1: The passenger clicks request message button on a pilot's page
- 2: The system checks to see if a message channel exists
- 3: the system sees a message channel does not exists and sends a request to the pilot
- 4: The passenger gets the success message "message request successfully sent"

Alternatives:

3a: the system sees a message channel exists

4a: The passenger get the message "If it does, it displays the message "message contact already exists, go to your messages to contact this pilot"

Test Situations:

- 1) A message channel does not exist
- 2) A message channel does exist
- 3) The success message gets displayed
- 4) The other message gets displayed

Solution:

The flow splits between 1 and 2. 3 follows 1 and 4 follows 2.

Test Coverage:

Base: number of main and alternative scenarios = 3

Test situations cover all 3 cases

100% coverage of use case

Title: Make Payment

Actors: Passenger and Pilot

Main Scenario:

- 1: The Passenger clicks make payment button
- 2: The system displays the payment page with the cost to be payed
- 3: The Passenger clicks the confirm payment button
- 4: The system routes money to the specified pilot and displays a confirmation
- 5: The Passenger sees the confirmation

Alternatives:

3a: The Passenger clicks the cancel button and the transaction is canceled

Test Situations:

- 1) The Passenger confirms the payment

2) The Passenger cancels the transaction

Solution:

For test situation 2, the click should trigger a different path where the transaction is canceled and the payment is still needed.

Test Coverage:

Base: number of main and alternative scenarios = 2

Test situations cover all 2 cases

100% coverage of use case

Title: Write a Review

Actors: Passenger and Pilot

Main Scenario:

- 1: The Passenger clicks write a review button
- 2: The system displays the review popup
- 3: The Passenger enters the binary and text review and clicks send
- 4: The system sends the review to the pilot and adds it to the pilot's reviews and sends a confirmation message to the Passenger
- 5: The Passenger gets the confirmation message

Alternatives:

4a: The system senses the binary is not entered so it does not send the review and asks the Passenger to enter the binary

4b: The system senses the text is not entered so it does not send the review and asks the Passenger to enter the text.

Test Situations:

- 1) The Passenger enter all the information
- 2) The Passenger does not enter the binary
- 3) The Passenger does not enter the text

Solution:

The system should check to make sure that the binary and text are entered for each review and if not the review is not sent or added to the pilot account and the Passenger gets a message on the UI that informs them of what needs to be entered.

Test Coverage:

Base: number of main and alternative scenarios = 3

Test situations cover all 3 cases

100% coverage of use case

Unit Testing (10-20, at least 5 implemented in code)

On the Login Page

1. Test the heading
 - a. Render without crashing
 - b. Render with "Firebase Auth Login"
2. Test the email

- a. Render without crashing
 - b. Render with placeholder "Email Address"
3. Test the password input
 - a. Render without crashing
 - b. Render with placeholder "password"
4. Test the Log In button
 - a. Render without crashing
 - b. Render with text Log In
5. Test the google button
 - a. Render without crashing
6. Test the link
 - a. Render without crashing
 - b. Link to "/signup"

On the Home Page

7. Test Log Out button
 - a. Render without crashing
 - b. Render with text "Log Out"
8. Test Firestore Demo button
 - a. Render without crashing
 - b. Render with text "Firestore Demo"
9. Test New Account button
 - a. Render without crashing
 - b. Render with text "New Account"
10. Test Messenger Test button
 - a. Render without crashing
 - b. Render with text "Messenger Test"
11. Test CreateConversation Test
 - a. Render without crashing
 - b. Render with text "CreateConversation Test"

To implement testing with code, many different approaches were attempted, but all were unsuccessful.

Testing For the Login Function

```
import React from "react";
import ReactDOM from "react-dom";
import Login from "../Login";
//import ShallowRenderer from "react-test-renderer/shallow";
import {render, screen, cleanup,renderIntoDocument} from
"@testing-library/react"

/*import {useState} from 'react';
import {Card, Button, Form, Alert} from 'react-bootstrap';
```

```

import '../..css/login.css';
import GoogleButton from 'react-google-button';
import {Link, useNavigate} from 'react-router-dom';
import {useUserAuth} from "../..contexts/UserAuthContext";*/
import TestUtils from "@vue/test-utils"

// test try number 8
afterEach(cleanup);
it("please work", function () {
  var component = TestUtils.renderIntoDocument(
    <Login />
  );
  var heading = TestUtils.findRenderedDOMComponentWithTag(
    component, "h2"
  );

  expect(heading.getDOMNode().textContent).toEqual("Firebase Auth
Login");
})

// test number 7
/*test("render something please", ()=>{
  render (<Login />);
  const heading = screen.getByTestId("heading");

  expect(heading).toBeInTheDocument();

}); */

// test number 6
/*
afterEach(cleanup);
test("rendering Login", () => {
  const {getByTestId} = render(<Login />);
  getByTestId("heading");
});*/

// test number 5
/*it("render without crashing", () =>{
  const renderer = new ShallowRenderer();

```

```

    renderer.render(<Login />);
    const result = renderer.getRenderOutput();

    expect(result.type).toBe("div");
  })*//

// the first couple tries got deleted when I pulled the Git repository

```

Testing for the Home Function

```

import React from "react";
import ReactDOM from "react-dom";
import Home from "../Home";
//import ShallowRenderer from "react-test-renderer/shallow";
import {render, screen, cleanup, renderIntoDocument} from
"@testing-library/react"
import TestUtils from "@vue/test-utils"

afterEach(cleanup);
it("please render something", function () {
  var component = TestUtils.renderIntoDocument(
    <Home />
  );
  var button1 = TestUtils.findRenderedDOMComponentWithTag(
    component, "Button"
  );
  var button2 = TestUtils.findRenderedDOMComponentWithTag(
    component, "Button"
  );
  var button3 = TestUtils.findRenderedDOMComponentWithTag(
    component, "Button"
  );
  var button4 = TestUtils.findRenderedDOMComponentWithTag(
    component, "Button"
  );
  var button5 = TestUtils.findRenderedDOMComponentWithTag(
    component, "Button"
  );
});

```

```
expect(button1.DOMNode().textContent).toEqual("Log Out");
expect(button2.DOMNode().textContent).toEqual("Firestore Demo");
expect(button3.DOMNode().textContent).toEqual("New Account");
expect(button4.DOMNode().textContent).toEqual("Messenger Test");
expect(button5.DOMNode().textContent).toEqual("CreateConversation
Test");
});

// other ways were tried with Home, but they were
// deleted why I pulled main from the Git repository
```

MANY (12+) hours were spent on looking into and researching as well as implementing testing using Jest and various add-ons.

```
at Object.<anonymous> (src/components/__tests__/Home.test.js)

Test Suites: 2 failed, 2 total
Tests:       2 failed, 2 total
Snapshots:   0 total
Time:        9.027 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
PS C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away>
```


The results were various reasons for failure, it is suspected that it has something to do with the methods defined within the Home.js and Login.js files.

```
console.error
Error: Uncaught [Error: useNavigate() may be used only in the context of a <Router> component.]
    at reportException (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\helpers\runtime-script-errors.js:66:24)
    at innerInvokeEventListeners (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\events\EventTarget-impl.js:341:9)
    at invokeEventListeners (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\events\EventTarget-impl.js:274:3)
    at HTMLUnknownElementImpl.dispatch (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\events\EventTarget-impl.js:221:9)
    at HTMLUnknownElementImpl.dispatchEvent (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\events\EventTarget-impl.js:94:17)
    at HTMLUnknownElementImpl.dispatchEvent (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\generated\EventTarget-impl.js:231:34)
    at Object.invokeGuardedCallbackDev (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:3994:16)
    at invokeGuardedCallback (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:4056:31)
    at beginWork$1 (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:23964:7)
    at performUnitOfWork (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:22779:12) Error: useNavigate() may be u
ed only in the context of a <Router> component.
    at invariant (C:\Users\Alison\Desktop\New folder\GreyMatter\packages\react-router\index.tsx:24:26)
    at useNavigate (C:\Users\Alison\Desktop\New folder\GreyMatter\packages\react-router\index.tsx:512:3)
    at Login (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\src\components\Login.js:13:19)
    at renderWithHooks (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:14985:18)
    at mountIndeterminateComponent (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:17811:13)
    at beginWork (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:19049:16)
    at HTMLUnknownElement.callCallback (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:3945:14)
    at HTMLUnknownElement.callTheUserObjectsOperation (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\generated\EventListener.js
26:30)
    at innerInvokeEventListeners (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\events\EventTarget-impl.js:338:25)
    at invokeEventListeners (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\events\EventTarget-impl.js:274:3)
    at HTMLUnknownElementImpl.dispatch (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\events\EventTarget-impl.js:221:9)
    at HTMLUnknownElementImpl.dispatchEvent (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\events\EventTarget-impl.js:94:17)
    at HTMLUnknownElementImpl.dispatchEvent (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\jsdom\lib\jsdom\living\generated\EventTarget-impl.js:231:34)
    at Object.invokeGuardedCallbackDev (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:3994:16)
    at invokeGuardedCallback (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:4056:31)
    at beginWork$1 (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:23964:7)
    at performUnitOfWork (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:22779:12)
    at workLoopSync (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:22707:5)
    at renderRootSync (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:22670:7)
    at performSyncWorkOnRoot (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:22293:18)
    at scheduleUpdateOnFiber (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:21881:7)
    at updateContainer (C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:25482:3)
    at C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away\node_modules\react-dom\cjs\react-dom.development.js:26031:7
```

```
FAIL src/components/__tests__/Home.test.js (5.076 s)
  • please render something

  TypeError: Cannot read properties of undefined (reading 'renderIntoDocument')

   9 |   afterEach(cleanup);
  10 |   it("please render something", function () {
> 11 |     var component = TestUtils.renderIntoDocument(
      |                               ^
  12 |       <Home />
  13 |     );
  14 |     var button1 = TestUtils.findRenderedDOMComponentWithTag(

at Object.<anonymous> (src/components/__tests__/Home.test.js:11:31)

Test Suites: 2 failed, 2 total
Tests:       2 failed, 2 total
Snapshots:   0 total
Time:        9.027 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
PS C:\Users\Alison\Desktop\New folder\GreyMatter\fly-away> []
```

These are two of the many error that occurred

The top picture occurred the most often

The second is the most recent

This code has not been committed to the Github repository due to the error within the code that reside on a local clone.

Acceptance Testing

At this time no acceptance testing has been done as the most unique features of the application are in progress or yet to be developed. Once more of the development has been completed such there are many working components like the Login page, the Signup page, the Home page, individual User Pager, a working search function, a fully functional booking and schedule system, a fully function messaging system and such, that would allow for a user to get an accurate representation of what the application is meant to be, will an acceptance test with potential users be run and documented.