Tutor Group

Test Plan

Table Of Contents:

1. Figures
2. Unit testing
3. Use case testing
   1. Log In
   2. Apply to Tutor
   3. Create Student Profile
   4. Post Listing
4. Acceptance testing

Figures

Text

Description automatically generated

Figure 1: ‘npm run test’ results

As shown in Image 1, testing has uncovered many issues in our application. These issues seem to stem mainly from the structure and navigation of the application. The components were also difficult to test as they are not completely developed. Our tests show that there is an error in the console and HTML creation of the application. There are also issues with many of the forms with submission, as post submission actions have not been implemented yet.

Unit Testing

*Testing Files:*

App.test.js

* UI testing
* Testing of toolbar functionality
* This test passed
* Testing function descriptions and success
  + “home screen rendered” —passed
  + “moved to tutor application” —failed
  + “moved to create profile” —failed
  + “moved to log in” —failed

LoginForm.test.js

* UI testing
* Form submission test
* Testing function descriptions and success
  + “login form rendered” —passed
  + “login form accepted after submit button” —failed

TutorApply.test.js

* UI testing
* Form submission test
* Testing function descriptions and success
  + ‘Tutor application rendered” —failed
  + “Tutor application accepted after submit button” —failed

homeScreen.test.js

* UI testing
* Button testing to see if the home screen buttons take you to a new page
* Testing function descriptions and success
  + “home screen rendered” —failed
  + “Home screen buttons work” —failed

postListing.test.js

* UI testing.
* Testing of the drop-down menu
* Testing function descriptions and success
  + “drop down select application rendered” —passed
  + “selections accepted after button press” —failed

studentProfile.test.js

* Testing function descriptions and success
  + “Student Profile rendered” —passed
  + “Student Profile form accepted after pressing submit” —passed

Use Case Testing

Title: Log In

Actors: User

Main Scenario:

1. User selects log in from tool bar
2. System displays log in page
3. User enters username and password into fields
4. User presses submit
5. System captures responses
6. System verifies username and password
7. System displays account home screen

Alternatives:

2a. System does not display log in page

2a1. System fails to log in user

3a. User cannot type in input fields

3a1 System fails to log in user

4a. User does not press submit

4a1. System fails to log in user

5a. System does not capture inputs

5a1. System fails to log in user

6a. System cannot access database

6a2. System fails to log in user

6b. User entered incorrect username and password

6b1. System fails to log in user

7a. System does not display account home screen

7a1. System fails to log in user

Test situation:

1. Tool bar selection does not reroute
2. System does not display application form
3. User cannot type in input fields
4. User does not press submit
5. System does not capture what is entered into input fields
6. System cannot access database
7. User entered incorrect username and password
8. System does not display account home screen

Test coverage:

100% coverage

Note: These test situations were not implemented. The event of logging in was tested with LoginForm.test.js



Title: Apply to Tutor

Actors: User

Main Scenario:

1. User selects “apply to tutor” from tool bar
2. System displays tutor application form
3. User enters information into form fields
4. User presses submit
5. System captures responses
6. System submits application to manager account
7. System displays submission verification

Alternatives:

2a. System does not display tutor application form

2a1. System fails to apply the tutor

3a. User cannot enter information in fields

3a1. System fails to apply the tutor

4a. User does not press submit button

4a1. System fails to apply the tutor

5a. System does not capture responses

5a1. System fails to apply the tutor

6a. System cannot access database

6a1. System fails to apply the tutor

6b. Manager account not updated

6b1. System fails to apply the tutor

7a. System does not display submission verification

7a1. System fails to apply the tutor

Test situation:

1. Tutor apply selection does not reroute
2. System does not display application form
3. User cannot type in input fields
4. User does not press submit
5. System does not capture information from input fields
6. System cannot access database
7. Manager account is not accessible

Test coverage:

100% coverage

Note: These test situations were not implemented. The event of Applying to tutor was tested with TutorApply.test.js

Title: Create Student Profile

Actors: User

Main Scenario:

1. User selects “sign up”
2. System displays create student profile form
3. User enters information into fields
4. User presses submit
5. System captures responses
6. System submits responses to database
7. System displays student profile page

Alternatives:

2a. System does not display form

2a1. System fails to create student profile

3a. User cannot enter information in fields

3a1. System fails to create student profile

4a. User does not press submit

4a1. System fails to create student profile

5a. System does not capture responses

5a1. System fails to create student profile

6a. System cannot access database

6a1. System fails to create student profile

7a. System does not display student profile page

7a1. System fails to create student profile

Test situation:

1. Student profile form is not rendered
2. User cannot type in input fields
3. User does not press submit
4. System doesn’t capture responses
5. System is offline and can’t access database
6. Student profile does not render

Test coverage:

100% coverage

Note: These test situations were not implemented. The event of creating a profile was tested with studentProfile.test.js

Title: Post Listing

Actors: Student Users

Main Scenario:

1. User selects “Post Listing Button”
2. System displays post listing page
3. User enters choices into the dropdown menu
4. User presses post listing
5. System captures responses
6. System stores the listing in the database

Alternatives:

2a. System doesn’t display the webpage/form

2a1. System fails to compile the webpage code

3a. User cannot select anything form the dropdown menu

3a1. System fails to create post listing

4a. User cannot click the final post listing button on the post listing webpage

4a1. System fails to create post listing

5a. Post listing data is not stored in the database

5a1. The post listing is create but is never stored so system looses the data.

Test situation:

1. Post listing button works
2. Dropdown menu for choices works
3. Post listing button on the post listing page does work
4. However the post listing data does not store the data in the database entered into post listing.

Test coverage:

100% coverage

Note: These test situations were not implemented. The event of posting a listing was tested with postListing.test.js

Acceptance Testing

Customer testing of the application showed that the application does not meet customer expectations. The customer who reviewed the application pointed out 3 main issues. The current navigation is not appropriate. One should not be able to post a listing or create a profile without an account. Forms such as tutor application and log in does not lead the user past the form. The application has not been completed in implementation and therefor does not pass acceptance testing. Restructuring of the application to follow previous diagrams outlined in sequence diagram document, software architecture document, and design pattern document is required.