

Open Freelance hub

Testing Report

Dean Lynch & David Weir

Table of Contents

- [1. Use Case Testing](#)
- [2. Additional Test Cases](#)
- [3. Smart Contract Unit Testing](#)
- [4. Heuristic Testing](#)
- [5. Real World Testing](#)
- [6. Issues Found](#)

Use Case Testing

Use Case No.	Title	Steps	Result
1	Posting a Job	1) Choose 'post a job' 2) Enter the details of a job 3) Click 'Post Job' & approve the transaction Expected Result: Job is posted and is visible on the job list page	PASS
2	Applying to a Job	1) Create an open job as specified in Test Case 1 2) Sign into a different account 3) Enter the job page of the previously created job 4) Click the 'apply' button and approve the transaction 5) Sign into the the account that owns the job 6) Enter the job page Expected Result: The address of the non-owner account is visible in the list of applicants	PASS

3	Accepting a candidate for a job	1) Apply to a job as in Test Case 2 2) As the owner of the job, click 'Accept' on an applicant Expected Result: The address of the candidate appears as the 'Accepted Worker'. The status of the job is marked as 'In Progress'.	PASS
4	Reviewing another party	1) Accept a worker as in Test Case 3 2) As the accepted worker, enter the job page 3) Click 'Confirm work completed' and complete a review of the employer 4) As the owner of the job, choose 'Complete Job' and complete a review of the worker Expected Result: Both reviews are completed. The job status is marked as closed. The payment amount is transferred to the worker. The reviews of each party can be seen on the job page.	PASS
5	Registering an ethereum account	1) Enter the site as an unregistered user 2) Choose 'Register Account' 3) Enter the details for the account 4) Click 'Register' and approve the transaction Expected Result: The ethereum account is registered. The user details can be seen in the list of users, which will link to an account page for the new users.	PASS

Additional Test Cases

Test Case No.	Title	Steps	Result
6	Completing a Job	1) Post a job and accept a worker for the job 2) As the accepted worker, enter the job page and click 'Confirm work completed' 3) Complete a review of the employer for the job 4) As the owner of the job, enter the job page and click 'Complete Job' 5) Complete a review for the worker Expected Result: The job is completed. The	PASS

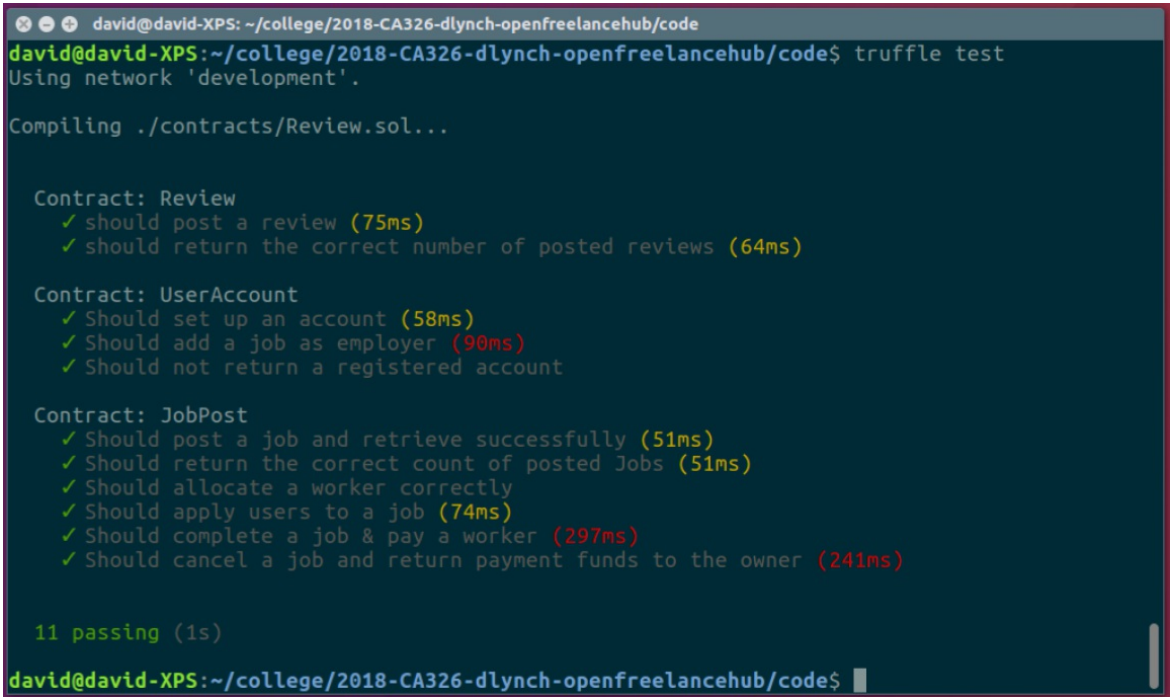
		status of the job is set to closed. The payment amount specified in the job post is tranferred to the etheruem account of the accepted worker.	
7	Searching for a Job	<p><i>Note:</i> The site must be populated with jobs for this test case.</p> <p>1) Enter the search keywords for a job into the search bar</p> <p>2) Click the 'Search' dropdown and selct 'search jobs'</p> <p>Expected Result: A search results page is generated with the result of the search. The correct job that was searched for and is posted correctly, is returned</p>	PASS
8	Searching for a User	<p><i>Note:</i> The site must be populated with users for this test case.</p> <p>1) Enter the search keywords for a user into the search bar</p> <p>2) Click the 'Search' dropdown and selct 'search users'</p> <p>Expected Result: A search results page is generated with the result of the search. The correct user that was searched for and is posted correctly, is returned</p>	PASS
9	Browsing Jobs	<p><i>Note:</i> The site must be populated with jobs for this test case.</p> <p>1) Choose 'Browse Jobs' from the home page</p> <p>2) Traverse the pages of the jobs</p> <p>Expected Result: All jobs are displayed in a number of different pages. They have different statuses: Open, In Progress, and Closed</p>	PASS
10	Canceling a Job	<p>1) Post a job and accept a worker for the job</p> <p>2) As the owner of the job, enter the job page</p> <p>3) Click 'Cancel Job' and approve the transaction</p> <p>Expected Result: The job status is marked as 'Closed' in the job list. The payment amount for the job is returned to the owners account.</p>	PASS

Smart Contract Unit Testing

The Truffle framework which we used in the development of this project allowed for the creation of unit tests written in JavaScript. We created a number of these while developing the different contracts used in the application. They can be viewed here: [test directory](#).

They cover all of the functions used in the smart contracts to provide the functionality of the application. Each function does not have it's own unit test, however every function is used in at least one function. For example, when testing whether or not a review is posted to the blockchain both *postReview()* and *getReview()* are tested when running the *Should post a review* unit test.

Unit Test Result



```
david@david-XPS: ~/college/2018-CA326-dlynch-openfreelancehub/code
david@david-XPS:~/college/2018-CA326-dlynch-openfreelancehub/code$ truffle test
Using network 'development'.

Compiling ./contracts/Review.sol...

Contract: Review
  ✓ should post a review (75ms)
  ✓ should return the correct number of posted reviews (64ms)

Contract: UserAccount
  ✓ Should set up an account (58ms)
  ✓ Should add a job as employer (90ms)
  ✓ Should not return a registered account

Contract: JobPost
  ✓ Should post a job and retrieve successfully (51ms)
  ✓ Should return the correct count of posted Jobs (51ms)
  ✓ Should allocate a worker correctly
  ✓ Should apply users to a job (74ms)
  ✓ Should complete a job & pay a worker (297ms)
  ✓ Should cancel a job and return payment funds to the owner (241ms)

11 passing (1s)
david@david-XPS:~/college/2018-CA326-dlynch-openfreelancehub/code$
```

These tests can be run using the following command in the code directory: *truffle test* (Note: *testrpc* must be running in a separate terminal window)

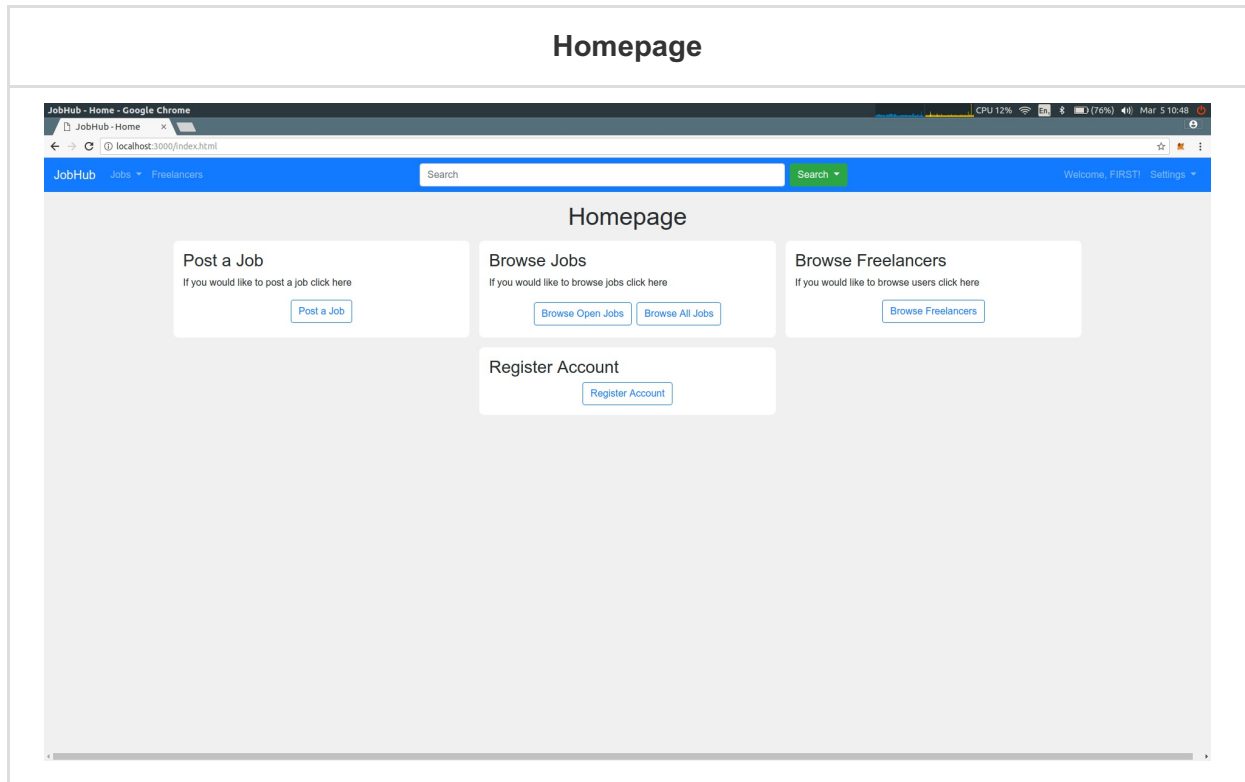
Heuristic Testing

Nielsen's Heuristics

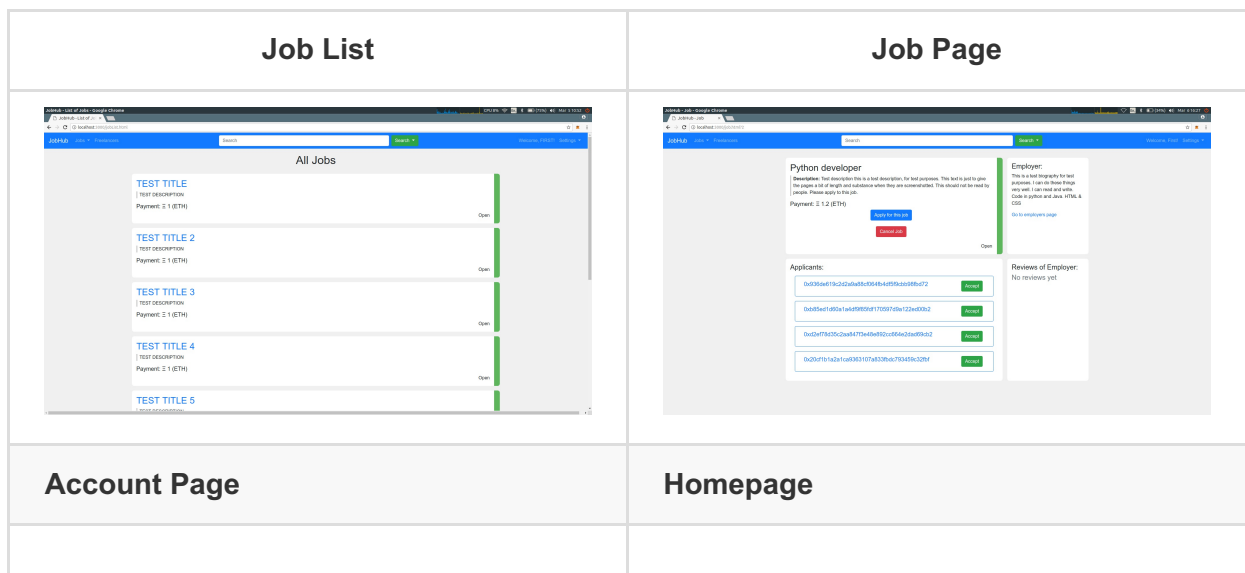
- **Simple & Natural Dialog:** All dialogue with the user on the site is kept to a minimum. Dialog in the form of popup's on the site is written in natural and logical language. General dialog such as that used on the homepage is also simple in order for users to simply figure out what they would like to do.
- **Speak the User's Language:** No complex language is used for the different dialogues in the application. However, some technological language/knowledge is unavoidable due to

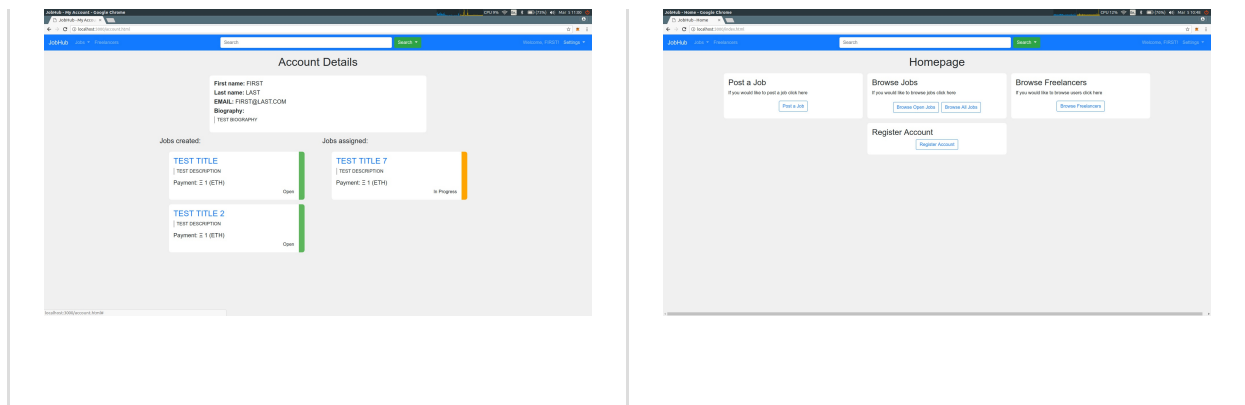
nature of blockchain applications and the fact that they are an emerging technology. This means that some knowledge of the blockchain is necessary when using any blockchain application.

- **Minimise the user's memory load:** The application functions in a very similar way to currently operating freelance sites, so any users of these sites would not find it difficult to begin using this application. All users will also be able to traverse the main functions of the through the homepage.

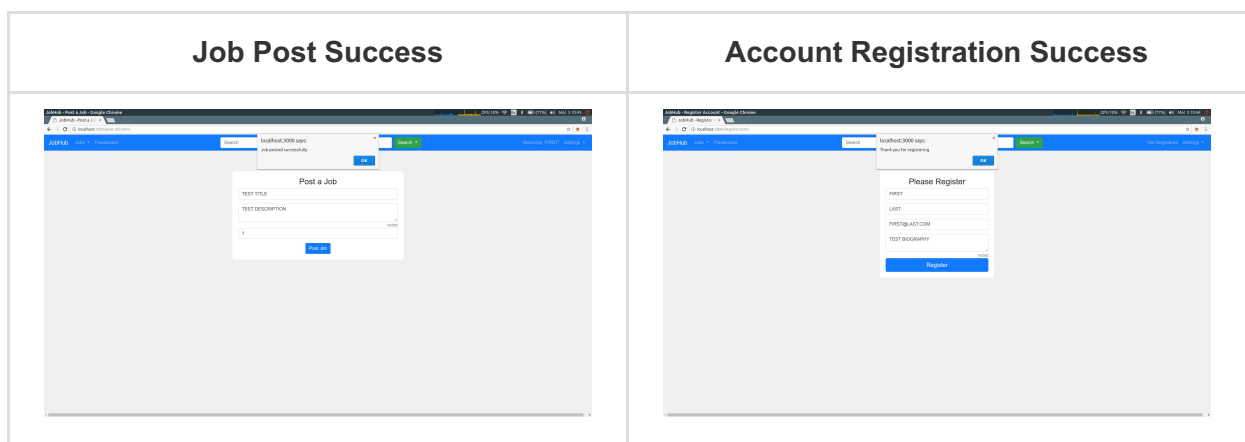


- **Consistency:** All language in the application is consistent in that it is kept simple as mentioned previously. The style of the site is also kept consistent as it keeps with the same theme throughout the application as shown in the screenshots below.

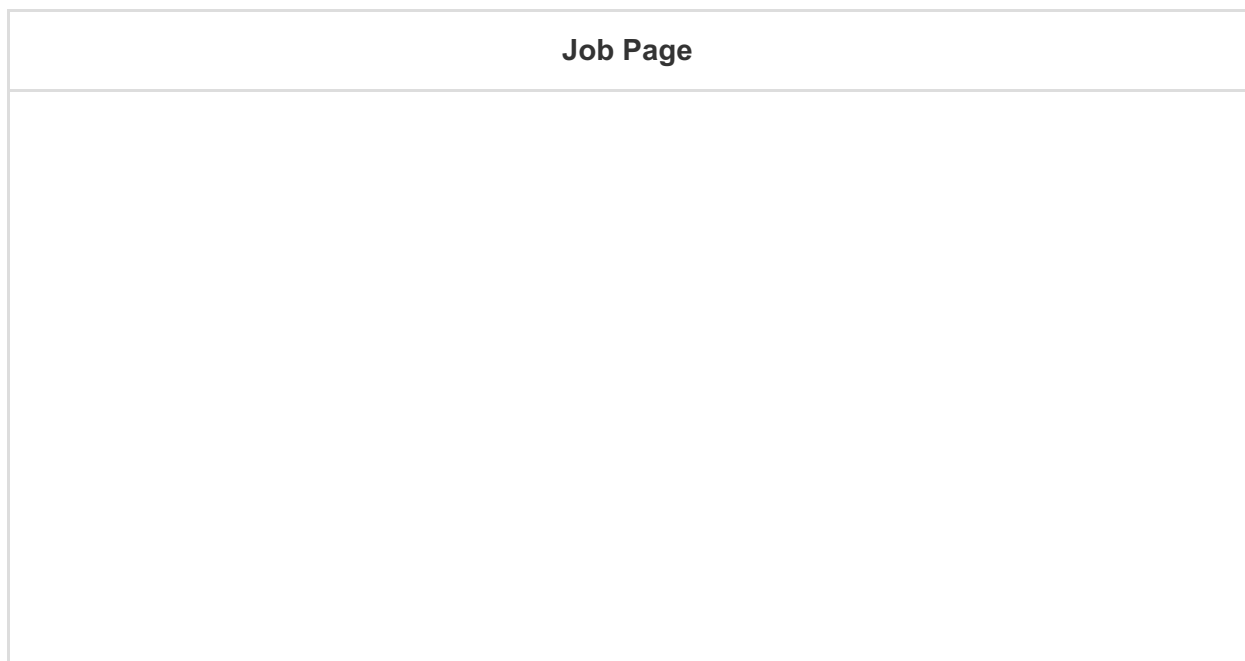


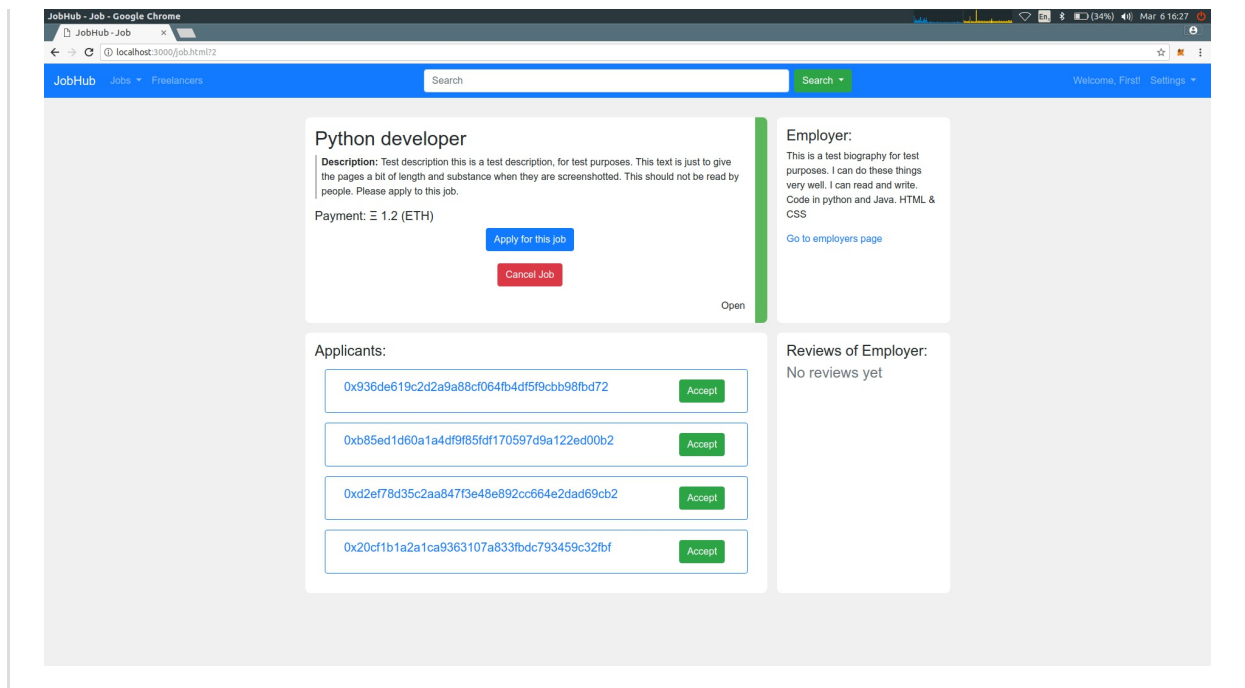


- **Feedback:** When the user completes an action on the site, such as posting a new job or applying to a job, a popup is displayed informing them of the success of their action.

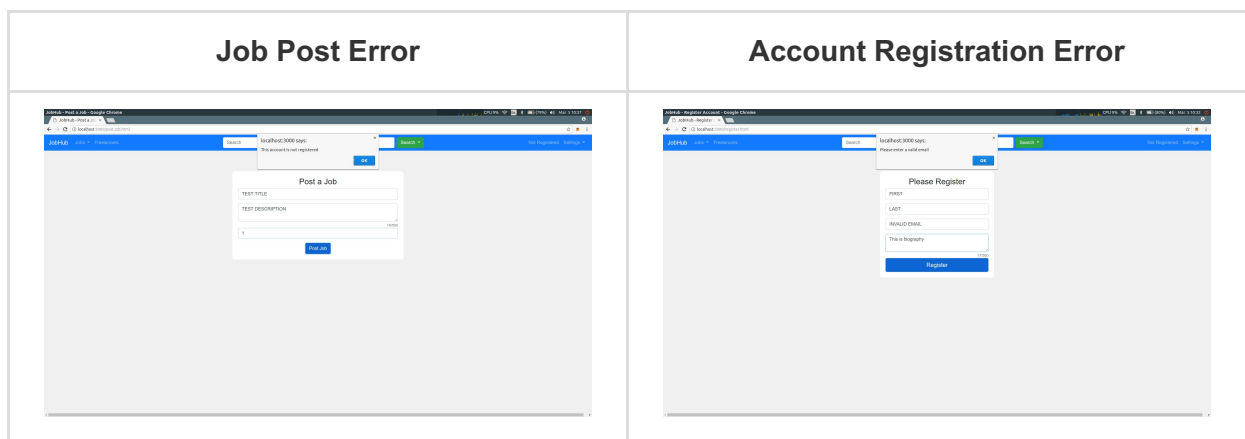


- **Clearly Marked Exits:** As this application browser based the user can simply use the back button if they have entered an incorrect page. If the user has posted a job by accident they can cancel the job which will return the payment amount of ether to the user, they will however be charged the gas fee to process this cancellation.

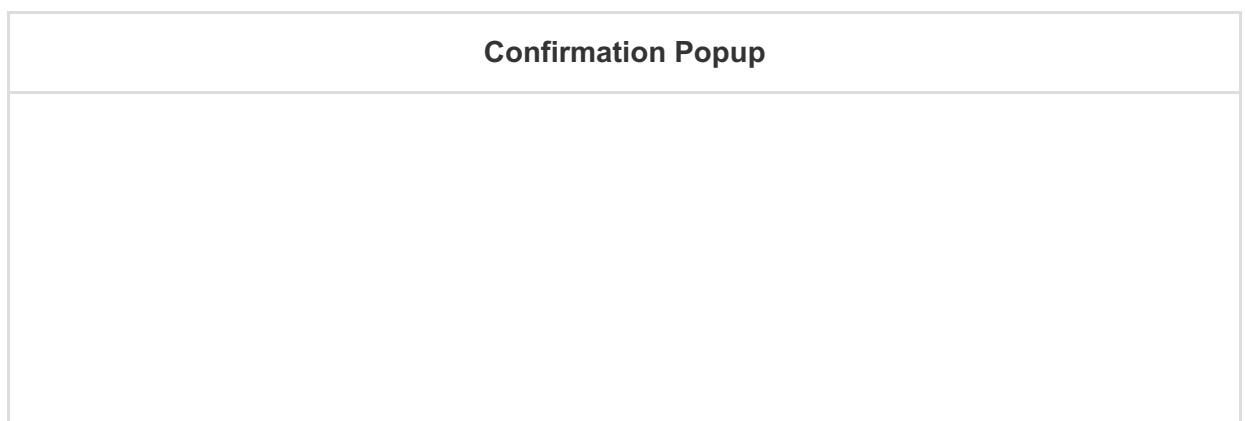


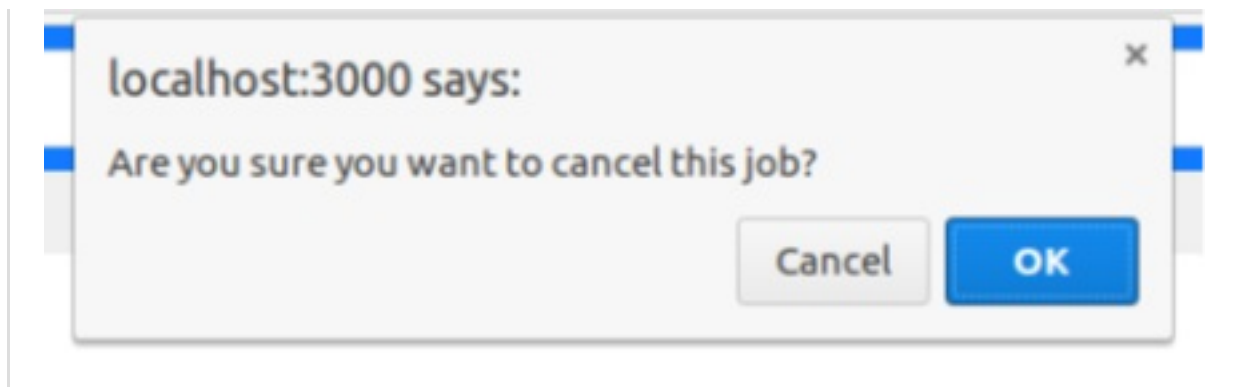


- **Good Error Messages:** When the user attempts to complete an action that is not allowed, they are informed as to why they are not allowed to compete it using a popup.



- **Prevent Errors:** When completing certain actions in the application, the user is asked to first confirm that they would like to complete it. This is in place in order to prevent users from cancelling jobs incorrectly etc...





- **Help and Documentation:** Users have access to the user manual for the application through the settings section of the navigation bar. This manual will walk them through the functionality of the application.

ADD SCREENSHOT OF SETTINGS DROPDOWN WITH MANUAL

Accessibility Testing

Sight Difficulties

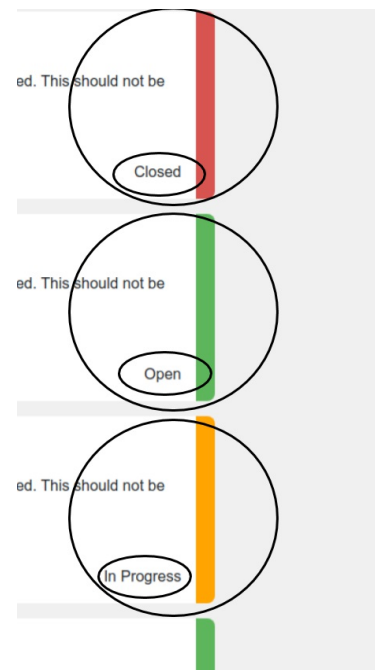
- The text sizes throughout the application are all of large font size to ensure that any users who have difficulties with their vision can read the information on the application with great ease.
- The colour scheme of the site was chosen as it is both aesthetically pleasing and it provides a high contrast to help users with colour blindness.
- The result of an online colour analysis tool states that the colour used in our links fail the contrast ratio for some users. To combat this we changed the colour of the links and underlined them when hovering over it.
- In order to combat some of the other colour contrast issues, such as the different colours for marking the job status we stated what the status of the job is beside the coloured status indicator.

** Analysis of Main Colours **	** Colour Contrast Fix **
Testing done on 49 elements	

Luminosity Contrast Ratio: 10 failures
Brightness difference: 0 failures
Color difference: 10 failures

full report only errors Click on the rows to test other colours

Node	Foreground	Background	Sample	Contrast Ratio	Brightness difference	Color difference
BODY	#212529	#F0F0F0	Sample Text	13.54:1 AAA	203	609
• A class="navbar-brand"	#007BFF	#F0F0F0	Sample Text	3.49:1	138	372
• SPAN class="navbar-toggler-icon"	#212529	#F0F0F0	Sample Text	13.54:1 AAA	203	609
• DIV id="navbarSupportedContent" class="collapse navbar-collapse"	#212529	#F0F0F0	Sample Text	13.54:1 AAA	203	609
•• UL class="navbar-nav mr-auto"	#212529	#F0F0F0	Sample Text	13.54:1 AAA	203	609
••• LI class="nav-item dropdown"	#212529	#F0F0F0	Sample Text	13.54:1 AAA	203	609
•••• A id="navbarDropdownMenuLink" class="nav-link dropdown-toggle"	#007BFF	#F0F0F0	Sample Text	3.49:1	138	372
••••• DIV class="dropdown-menu"	#212529	#FFFFFF	Sample Text	15.43:1 AAA	218	654
•••••• A class="dropdown-item"	#212529	#FFFFFF	Sample Text	15.43:1 AAA	218	654
••••••• A class="dropdown-item"	#212529	#FFFFFF	Sample Text	15.43:1 AAA	218	654



Motor Skill Difficulties

- The buttons etc... used in the application are kept large.
- The dropdowns on the site do not disappear if the mouse moves off the dropdown as is the case on some sites. This can cause issues for users with parkinsons etc...

Real World Testing (User Testing)

Test Plan

Each of the users who participated in our user testing were provided with a list of tasks and asked to complete these tasks before proceeding to delve freely into the site. The set of tasks cover the major functionalities of the application. The set tasks were as follows:

- Register an unregistered etheruem account.
- Post a job.
- Apply to a job.
- Accept an applicant for the job you have posted.
- Complete a job.
- Search for a job or user.

User Test Outcome

---NEED SCREENSHOTS OF FORM---

Issues Found

