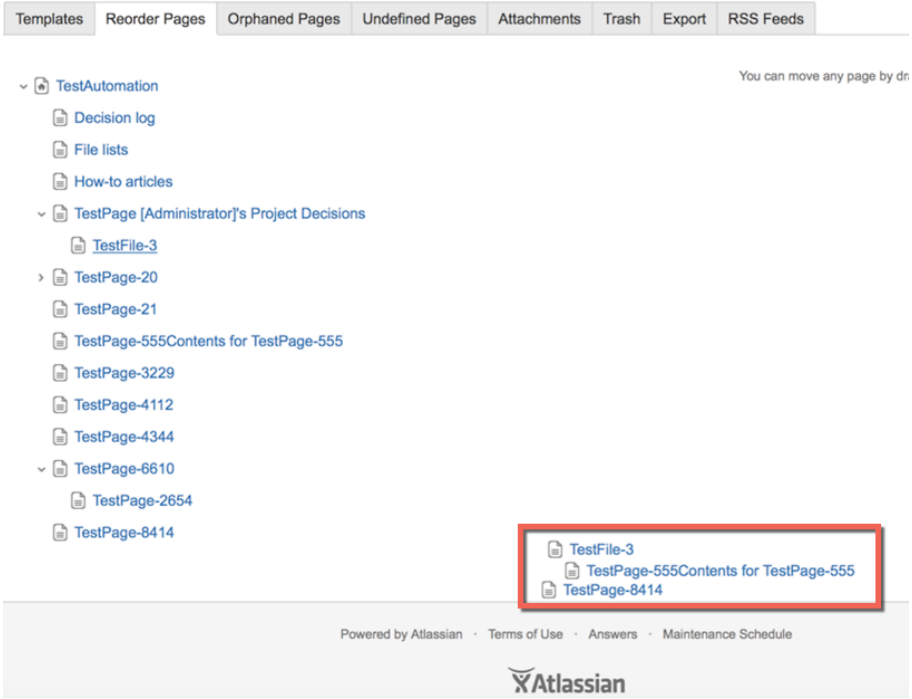
Exploratory Testing Exercise – Atlassian

- **A short description (1-2 sentences) of the approach you took to your testing.**

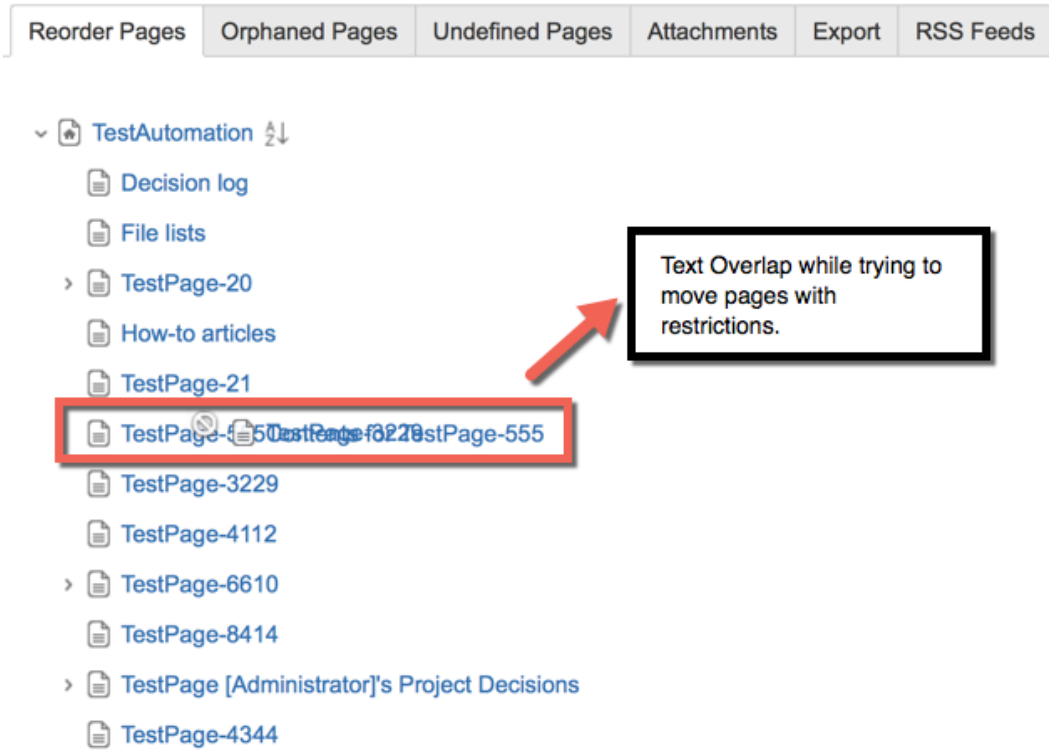
  I tried to approach the challenge of testing from all angles (end-user, business, security). Apart from the obvious defects in the application, I tried to find some subtle variances in behavior that could be a potential defect.

- **A bulleted list of the scenarios you tried, even if they were successful.**

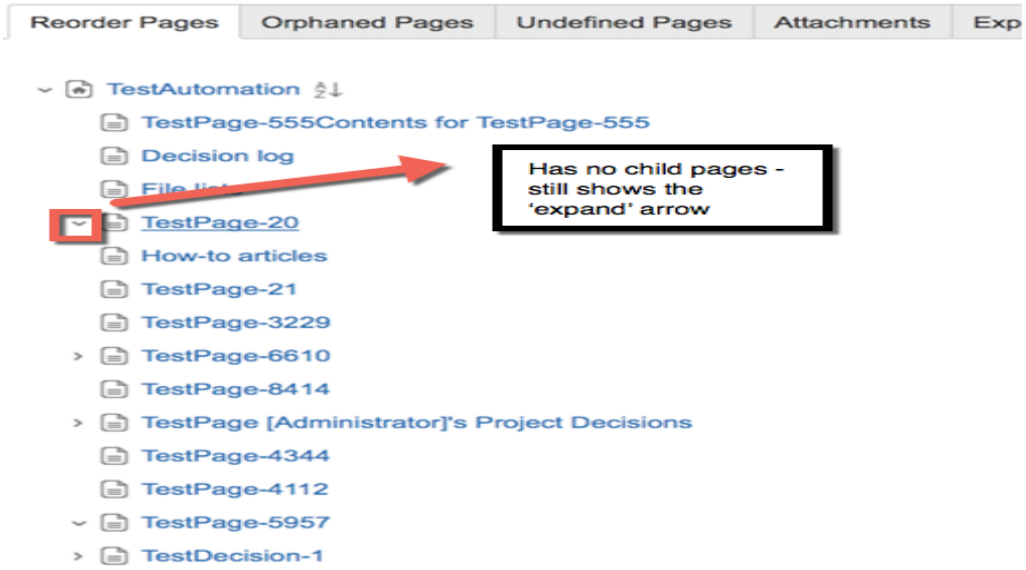
  ➢ Tried to move a child page outside the tree hierarchy container.
  Issue: Copy of the dragged item is set at various places on the page



  ➢ Tried reordering of restricted pages.
  Scenario: User 1 (Rajesh Kalloor [Administrator]) has given only view permissions to User 2(Rajesh Kalloor Test) to page 'TestPage-3229'. User2 logs in and tries to reorder pages
  Issue: User2 tried to reorder 'TestPage-3229', it does not move but the text over-laps with the existing structure of the tree. Moving unrestricted pages, however, works as expected.



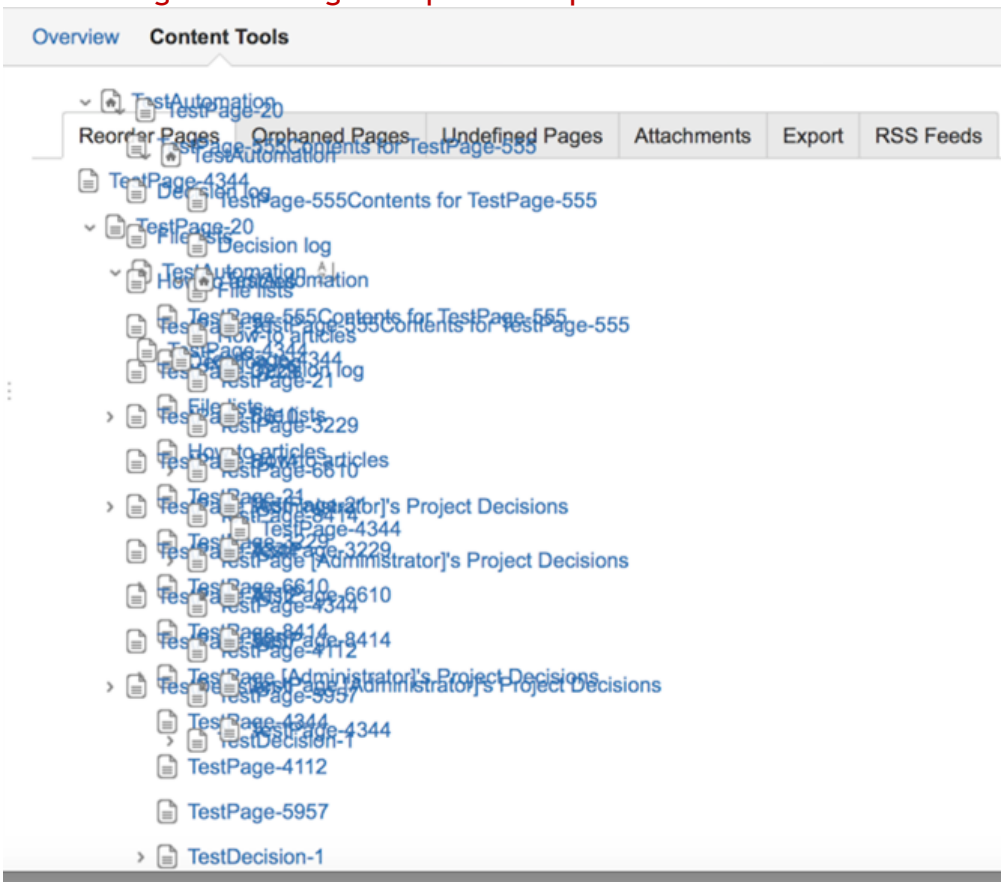
➢ Tried dragging and dropping parent page to its child pages-multiple times.
Issue: We see that pages without any children pages show the 'expand' arrow (though it does not work)



➢ Made a child page the Parent of its Parent, there is no way to undo this change.

Issue: No undo Operation after reordering of pages.

➢ Tried to move a large sub-tree (page with many child pages) on top of its parent page.
Issue: There is a lot of text overlapping. I am able to reorder pages over and over again creating multiple overlaps.



➢ Tried to reorder different pages and refresh, check if the order of pages was maintained after refresh.

➢ Tried to create a loop, by making the child page as the parent of its parent page. It throws an error which is expected.

➢ User 2: Rajesh Kalloor Test has only view permissions to page ‘TestPage-3229’.
Issue: User 2 is able to add child pages to this page (through drag and drop) as well as create new pages under this page. Since he has only view permissions to this page, any edit operations (including creating child pages) must be restricted.

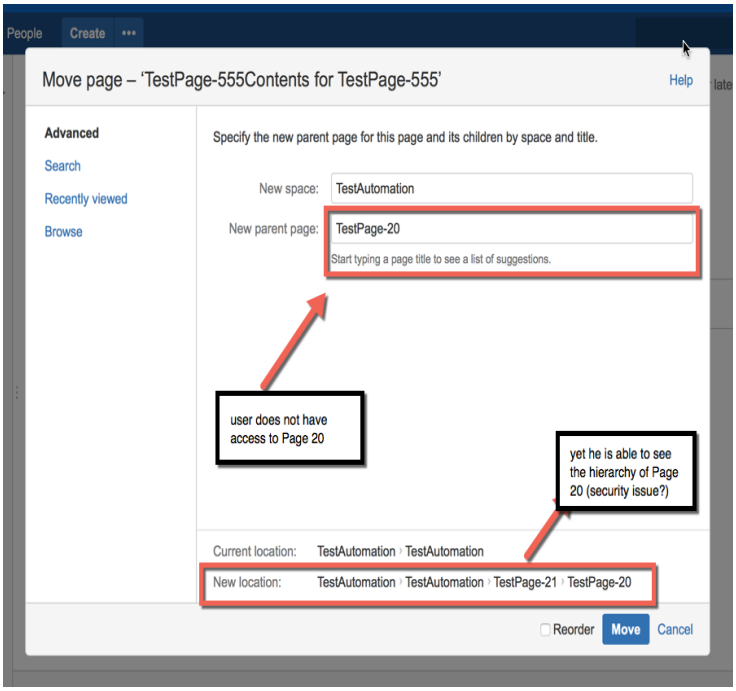
➢ Tried deleting children nodes, and see if the pages were getting deleted.

- Tried deleting a parent page with several child pages, the child pages were being adopted by the parent page's parent node or if the deleted node had no parent, the child page would become the parent.

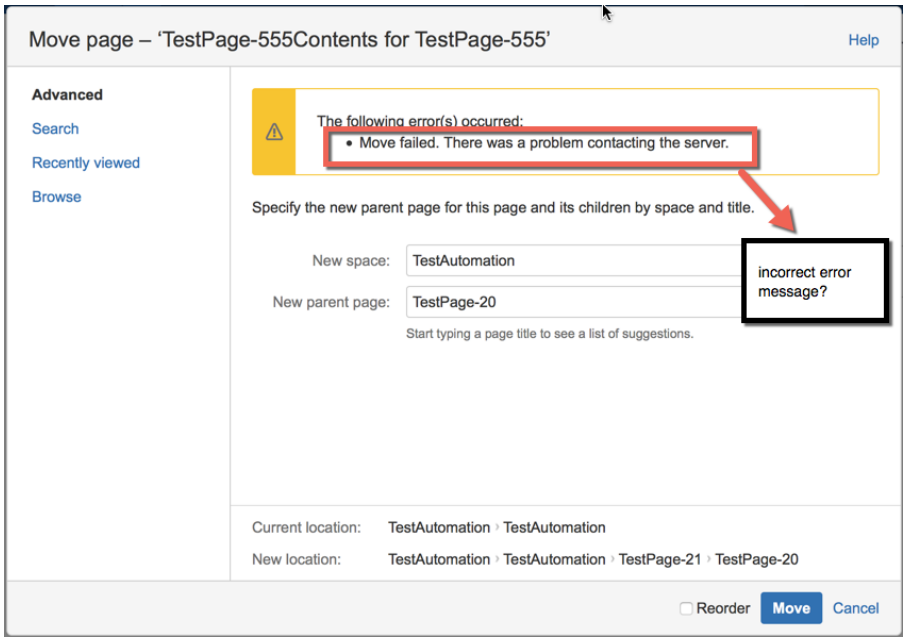
- Restrict User 2 [Rajesh Kalloor Test], from viewing page 'TestPage-20' which has many child pages. The User 2 should not be able to view the child pages of page 'TestPage-20'.

- Rename a page to the same name as another page in the same tree hierarchy. This throws an error pop-up message which is expected.

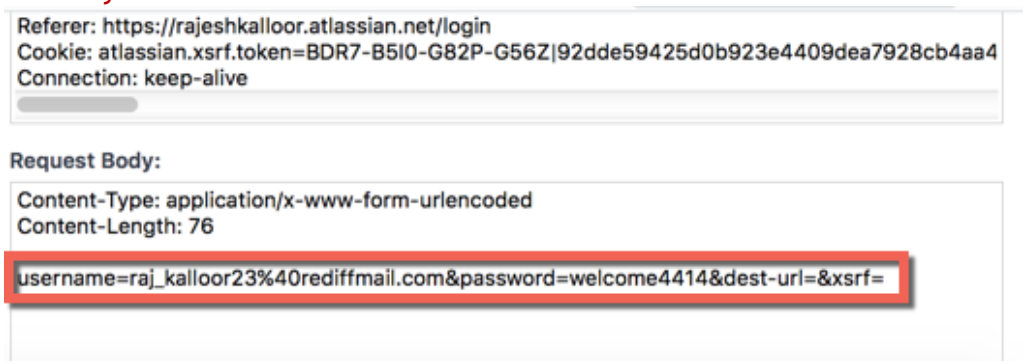
- User 2: Rajesh Kalloor Test has no access to Page – 'TestPage-20'.
  Issue 1: User 2 is able to go to a new page and change the parent of the page to 'TestPage-20' (to which he has no access). When he presses Enter key, he is able to see the path of 'TestPage-20' (as shown below)



Issue 2: Now when User 2 presses the Move button, he is shown an incorrect error message as shown below. The error message should tell him that he does not have the correct permissions to perform the operation.



➢ Additionally, while logging in as User 2: Rajesh Kalloor Test, I found that we are able to see the username and password in the request being sent to the server.
Issue: The Username and password are not encrypted. This is a potential security issue.



- **A quick description of any bugs you found.**

  Issues listed above in Red.

- **A list of further testing areas that you would want completed before you would be comfortable shipping this feature to millions of users. Consider the risks of the feature as you understand it.**

  1. We can further add browser compatibility tests and test across different platforms

2. We can have performance tests, in case of large tree structures – reordering pages should not take a lot of time, scrolling on the tree hierarchy should not result in page being stuck. Also extremely large trees with millions of child pages – can be paginated and reordering across different page must work.

3. We can test reordering across different spaces (both with and without restrictions)

4. Testing with multiple sessions - say - editing from 2 different sources like mobile and laptop and ensuring that the changes get saved. Also handling of cases where multiple users are trying to reorder the pages concurrently.

5. API tests for the backend server to ensure that the contracts are adhered to and all corner case validations are tested out.

6. DB level testing to ensure that the reordering is properly saved into the data stores and not just served out caches.

7. High availability testing to ensure that the reordering functionality is transparent to the user to some extent even if the different components fail behind the scenes - say - web server, database, downstream system etc.