## CS163 Test Plan

**Develop the test plan:** *For each member function that you plan to write, think about how to test it – what flow of control exists in the member function and how would you test out all conditions:*

|  |  |  |
| --- | --- | --- |
| **Test Cases for Add News** | **Expected Result** | **Verified?**  **(yes/no)** |
| No news stories (empty list) | News story should be added to head and both next pointers set appropriately to null. |  |
| 1 news story in list | Both pointers should point to appropriate node, and it should either be the new last node or head should point to it (check that the existing data hasn’t gone). |  |
| Multiple news stories in list | Check to make sure that new story has sorted into correct position by displaying, and check end cases. |  |
| Check return value | Should report correct number (0 or 1) based on situation. |  |

|  |  |  |
| --- | --- | --- |
| **Test Cases for Update News** | **Expected Result** | **Verified?**  **(yes/no)** |
| No news stories (empty list) | Exit, and check return value |  |
| 1 news story in list | Only this news story should display. Check to make sure that the number associated to this is correct. |  |
| Multiple news stories in list | The numbering of the stories should be correct. |  |
| After removing a story | Check that the numbers have accurately changed. |  |
| **Test Cases for Topic of Interest** | **Expected Result** | **Verified?**  **(yes/no)** |
| Enter different topic than existing topics | Report invalid topic THROUGH RETURN of 0. Should not display within function. |  |
| Don’t enter topic | Should return 0 and exit to main. |  |
| Enter valid topic | Should match and assign topic to variable, then return. |  |

|  |  |  |
| --- | --- | --- |
| **Test Cases for Display functions** | **Expected Result** | **Verified?**  **(yes/no)** |
| Enter for last placement (old time or uninterested topic) | Added to end of list, and set next to null. |  |
| Enter for first placement (newest time or interested topic) | Added to beginning of list, assign temp node to existing first story, then point head to new story. Should not delete rest of the list!! |  |
| Empty list | Display nothing |  |
| 1 story in list | Display just 1 |  |
| Display after removing an item | Should update without removed item |  |

|  |  |  |
| --- | --- | --- |
| **Test Cases for Remove Older** | **Expected Result** | **Verified?**  **(yes/no)** |
| Empty list | Remove nothing. |  |
| No stories older than the list | Remove nothing. Check that end still points to null. |  |
| 1 story older than the timestamp | Should remove, and end points to null. |  |
| Multiple older stories | Remove stories and end should still point to null. |  |
| All stories are older than the timestamp | Entire list should be deleted, and head points to null. |  |

|  |  |  |
| --- | --- | --- |
| **Test Cases for Favorite Story** | **Expected Result** | **Verified?**  **(yes/no)** |
| No news stories (empty list) | Exit, and should return 0 for no favorite selected. Check variable value. |  |
| 1 news story in list | Only this news story should display. Check to make sure that the number associated to this is correct. |  |
| Multiple news stories in list | The numbering of the stories should be correct. |  |
| After removing a story | Check that the numbers have accurately changed. |  |
| Favorite selected matches existing topics | “Favorite” variable should be same as topic |  |
| Favorite selected does not match existing topics | Return 0, and no favorite should be selected. |  |