## 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 Case(s)** | **Expected Result** | **Verified?**  **(yes/no)** |
| **Create venue- insert node of an already created entry** | **Return once string compare confirms match** |  |
| **Create venue - user inserts into an empty tree** | **Create new tree** |  |
| **Display by meal - the meal the user searched for deos not exist** | **Traverse through the bst and return a false boolean value to signal no meal match** |  |
| **Insert – user creates new data entry** | **String compare and traverse to past a leaf** |  |
| **Retrieve - retrieve a non-existing entry** | **Traverse through the hash table and return a false boolean value to signal no meal match** |  |
| **Remove by meal – remove from an empty tree** | **Check to see if there are entries and a zero to the calling routine to signal no table** |  |
| **Get user input – user tries skip input** | **Keep asking user until an entry is made** |  |
| **Get height – user tries to get height of an empty tree** | **Return zero to the calling routine** |  |
| **Is efficient – user tries to check if an empty tree is balanced** | **Return false to the calling routine** |  |
| **Display all – user tries to display and empty tree** | **Return zero to the calling routine to signal tree is empty** |  |

**Verify correctness:** Using the above test plan, create a test program that testhe interactions of all functions together.