Website Test Specification (WTS):

Version: 0.2

Date: 28/10/2019

# **How to use this document:**

* Write the requirement of the website function based on a story.
* Describe this requirement and the specifics of how it is supposed to be implemented.
* Write tests to make sure this requirement is met and list their numbers in the relevant test numbers section of the requirements table.
* If carrying out a review then make sure you add this document and a description of which tests the pull request passed / failed into the pull request approval / comment.
* If this document is updated to account for new features the following should occur:
  + 1. Before anything else go to the review section of word and click “Track Changes”
    2. Increment the version number by 0.1
    3. Update the date to the latest one
    4. The document should be added to the commit for whatever changes you are making ()

# **Requirements:**

|  |  |  |
| --- | --- | --- |
| **Name:** | **Description:** | **Relevant Test Number(s):** |
| **Searching for injury** | **As a User I want to be able to search for my injury.** | **3,6** |
| **Implement Map API** | **As a User I want to be able to see the locations of my search results on a map.** | **1, 2, 4, 5** |
| **Sorting Results** | **As a user I would like to be able to sort my results by multiple different criteria such as cost, state, zipcode, and distance.** | **7** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# **Tests:**

### 1. Map Marker Test:

1. Start-up the local server in your developer environment. (Eg. npm start in the NodeServer folder)
2. Start up the angular website in your developer environment making sure that the version you are using is trying to connect to the localhost address of the node server instead of the hosted one. (Eg. ng serve –o in the Website folder)
3. Search an operation code in the procedure search bar (Eg. 065 or 039 are known working codes).
4. **VERIFY** that the map displays **10** markers on the map underneath the table.

### 2. Map Display Test:

1. Start-up the local server in your developer environment. (Eg. npm start in the NodeServer folder)
2. Start up the angular website in your developer environment making sure that the version you are using is trying to connect to the localhost address of the node server instead of the hosted one. (Eg. ng serve –o in the Website folder)
3. **VERIFY** that the google maps API map at the bottom of the page displays and should be set to be looking over Africa by default in roadmap view.

### 3. Operation Code Search Test:

1. Start-up the local server in your developer environment. (Eg. npm start in the NodeServer folder)
2. Start up the angular website in your developer environment making sure that the version you are using is trying to connect to the localhost address of the node server instead of the hosted one. (Eg. ng serve –o in the Website folder)
3. **VERIFY** that the procedure displayed correctly relates to the code entered.
4. **VERIFY** that the table displays 100 results ordered in pages of 10.
5. **VERIFY** that the table can be paged through and each page shows unique results.

### 4. Map Marker Information Test:

1. Start-up the local server in your developer environment. (Eg. npm start in the NodeServer folder)
2. Start up the angular website in your developer environment making sure that the version you are using is trying to connect to the localhost address of the node server instead of the hosted one. (Eg. ng serve –o in the Website folder)
3. Search a medical procedure code in the procedure search bar (Eg. 065 & 039 are known working codes).
4. **VERIFY** that the map adds a marker for each result shown in the current page of the table.
5. **VERIFY** that each marker is clickable and on click displays the following: Hospital name, price, distance, and address.

### 5. Map Pagination Test:

1. Start-up the local server in your developer environment. (Eg. npm start in the NodeServer folder)
2. Start up the angular website in your developer environment making sure that the version you are using is trying to connect to the localhost address of the node server instead of the hosted one. (Eg. ng serve –o in the Website folder)
3. Search for a medical code in the procedure search bar (Eg. 065 & 039 are known working codes).
4. **VERIFY** that the table shows multiple pages with 10 items per page.
5. **VERIFY** that the table can be paged through and as it changes pages so too does the maps markers update.

### 6. Search Sorting (Min & Max) Test:

1. Start-up the local server in your developer environment. (Eg. npm start in the NodeServer folder)
2. Start up the angular website in your developer environment making sure that the version you are using is trying to connect to the localhost address of the node server instead of the hosted one. (Eg. ng serve –o in the Website folder)
3. Search for a medical code in the procedure search bar (Eg. 065 & 039 are known working codes) and add a minimum price of 100 and maximum price of 10000.
4. **VERIFY** that the table shows multiple pages with 10 items per page.
5. **VERIFY** that the table can be paged through and as it changes pages so too does the maps markers update.
6. **VERIFY** that the table does not have results that are greater than 10,000 in cost or less than 100.

### 7. Autocomplete Test:

1. Start-up the local server in your developer environment. (Eg. npm start in the NodeServer folder)
2. Start up the angular website in your developer environment making sure that the version you are using is trying to connect to the localhost address of the node server instead of the hosted one. (Eg. ng serve –o in the Website folder)
3. Search for a medical code in the procedure search bar (Eg. 065 & 039 are known working codes).
4. **VERIFY** that when entering into the location field it autocompletes the address automatically.
5. **VERIFY** that when entering into the procedure field it autocompletes the procedure name automatically when you are search for either the code or the name.