Change request log

# Concept Location

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We ran the system |  |
| 2 | We interacted with the system: after logging in we clicked on the urgent button at the top. | There was a flashing urgent button, so just wanted to check if there were any important notifications related to the fix. |
| 3 | We looked through the landing page that contains the watch list and clicked on various tabs until finding the history tab. | We saw that values in history were also needed to be fixed, so we clicked around to get familiar with what each button does and to also find the history tab to fix. |
| 4 | We opened our IDE and searched for watchlist. | Knew that the component we needed to fix was watch list but got over 600 results for it. |
| 5 | We did inspect element on the value on the site. | Wanted to know the code that displays the value onto the screen |
| 6 | Searched for the div code that we got from the previous step on the IDE | Wanted to know the classes that a div with the exact same configurations was present in, and there was only 1 class – watchlist.jsp |
| 7 | We inspected the file watchList.jsp | The div that produced value was in this class, so wanted to check it out |
| 8 | We searched for WatchList on our IDE and found/inspected the Java class WatchListDwr | This was the class associated with the WatchList, so we checked this to see if we needed to modify it. |
| 9 | We marked the Java class WatchListDw as “located” | We realized that this class had the methods related to points displayed in the home page |

**Time spent (in minutes):** 30

Classes and methods inspected:

* mangoSource/src/com/serotonin/mango/web/dwr/WatchListDwr.java
  + WatchListState createWatchListState(HttpServletRequest, DataPointVO, RuntimeManager, Map<String, Object>, User)
  + WatchListState addToWatchList(int)
  + List<WatchListState> getPointData()
  + Map<String, Object> init()

# Impact Analysis

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We visited the WatchListDwr class | This was where the functions to modify the points were located. |
| 2 | We checked where each of the methods listed above were called. | To check if those functions made an impact in any other class; found out that they do not |
| 3 | We inspected the DataPointDetailsDwr Java class. | We realized that this had to be changed because it was related to the history of the individual data point, which also had decimals. We found this through the same inspect element strategy mentioned in step 1. |
| 4 | We inspected the getStatsChartData method in DataPointDetailsDwr | We realized that this was related to the statistics data displayed, and that had decimal values to be fixed |
| 5 | We inspected the StatisticsChartRenderer class | We realized that this class was responsible for rendering the statistics in the getStatsChartData method. |
| 6 | We finalized the DataPointDetailsDwr, WatchListDwr, and StatisticsChartRenderer as the finalized classes to change. | We realized that these were the three classes where decimals needed to be fixed. |

**Time spent (in minutes):** 23

* mangoSource/src/com/serotonin/mango/web/dwr/DataPointDetailsDwr.java
  + DwrResponseI18n getHistoryTableData(int)
  + WatchListState getPointData()
  + DwrResponseI18n getStatsChartData(int, int, boolean)
* mangoSource/src/com/serotonin/mango/view/chart/StatisticsChartRenderer.java
  + void addDataToModel(Map<String, Object>, DataPointVO)

# Actualization

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | Set the PointValueTime value in getHistoryTableData of the DataPointDetailsDwr class with Math.floor(val \* 100)/100 | This expression truncates a decimal to 2 decimal places in the history portion of a data point detail |
| 2 | Set the PointValueTime value in getPointData of the DataPointDetailsDwr class to 2 decimal places using the same method as step 1 | This was to make sure that the first section that displayed value, time, and xid had the value truncated to 2 decimal places. |
| 3 | Set the PointValueTime value in the createWatchListState method of WatchListDwr to 2 decimal places using the same method as step 1 | This was to set the values of all points on the home screen to max 2 decimal places. |
| 4 | Updated the StartsAndRunTimeList object double fields in the addDataToModel method of the StatisticsChartRenderer class to 2 decimal places | This was to set the minimum, maximum, average, and sum in the statistics to 2 decimal places. |

**Time spent (in minutes):** 60

Inspected:

* mangoSource/src/com/serotonin/mango/view/chart/StatisticsChartRenderer.java
  + void addDataToModel(Map<String, Object>, DataPointVO
* mangoSource/src/com/serotonin/mango/web/dwr/DataPointDetailsDwr.java
  + DwrResponseI18n getHistoryTableData(int)
  + WatchListState getPointData()
  + DwrResponseI18n getStatsChartData(int, int, boolean)
* mangoSource/src/com/serotonin/mango/web/dwr/WatchListDwr.java
  + WatchListState createWatchListState(HttpServletRequest, DataPointVO, RuntimeManager, Map<String, Object>, User)
  + List<WatchListState> getPointData()

Changed:

* mangoSource/src/com/serotonin/mango/view/chart/StatisticsChartRenderer.java
  + void addDataToModel(Map<String, Object>, DataPointVO
* mangoSource/src/com/serotonin/mango/web/dwr/DataPointDetailsDwr.java
  + DwrResponseI18n getHistoryTableData(int)
  + WatchListState getPointData()
* mangoSource/src/com/serotonin/mango/web/dwr/WatchListDwr.java
  + WatchListState createWatchListState(HttpServletRequest, DataPointVO, RuntimeManager, Map<String, Object>, User)

# Validation

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | Test case defined: Values on the home page for all points should be 2 decimal places  Expected output: Values are truncated to 2 decimal places | This is the regular expected behavior.  The test passed. |
| 2 | Test case defined: Values in the history of data point details should be 2 decimal places  Expected output: Values are truncated to 2 decimal places | This is the regular expected behavior.  The test passed. |
| 3 | Test case defined: Value in the data summary (with value, time, xid) of data point details should be 2 decimal places  Expected output: Values are truncated to 2 decimal places | This is the regular expected behavior.  The test passed. |
| 4 | Test case defined: Minimum, maximum, average, and sum statistics of the statistics section of data point details should be 2 decimal places  Expected output: Values are truncated to 2 decimal places | This is the regular expected behavior.  The test passed. |
| 5 | Test case defined: Error shouldn’t come up when value does not exist  Expected output: Value is shown as n/a or blank everywhere | This is an edge case behavior  The test passed. |

**Time spent (in minutes):** 10

# Summary of the change request

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Time  (minutes) | No. of classes inspected | No. of classes changed | No. of methods inspected | No. of methods changes |
| Concept location | 30 | 1 | 0 | 4 | 0 |
| Impact Analysis | 23 | 2 | 0 | 4 | 0 |
| Actualization | 60 | 3 | 3 | 6 | 4 |
| Verification | 10 | 3 | 0 | 3 | 0 |
| Total | 123 | 9 | 3 | 17 | 4 |

# Conclusions

For this change, concept location was not too hard because the WatchList.java file was pretty easy to locate as the first class that needed to be changed (the watch list was on the home page, so that file needed to be edited). The impact analysis was also not too hard because there were only a few methods that I needed to check with respect to other components of the application being affected, and these methods were unrelated to the other components of the site. Moreover, finding the history class and then the stats class through that was relatively easy too. Actualization was the hardest part of this change for me because it took a while to find the methods that needed to be changed. While I inspected a lot of methods, it took a bit of time to see which methods in the specific classes had to be changed. So, it took some time to fix the issue present. Finally, verification took the least amount of time because I just had to monitor the different components of the application where decimal values were shown, and this was done with just a few button clicks.