# Mango-cr#1-TEAM-08 - Change request log

## 1. Concept Location

| Step # | Description | Rationale |
|---|---|---|
| 1 | *We ran the system* | |
| 2 | *We interacted with the system: after logging in we entered the schedule screen.* | *To get familiar with some of the features of the system, and identify the screens or graphical elements we had to change.* |
| 3 | *We performed a workspace search using the "watchlist" keyword using the Ctrl+H shortcut.* | *We aimed to gather additional insights into the origin and usage of the watchlist feature visible on the screen.* |
| 4 | *We went to the WatchListDwr.java file. We understood that the data is being fetched from WatchListDao class.* | *We understood the dependency of the data operations, thus navigated here.* |
| 5 | *Upon entering the WatchListDao class, we encountered the populateWatchListData method.* | *Given its name, it is reasonable to infer that this method is responsible for populating the data utilized in the watchlist feature.* |
| 6 | *From here, we went to DataPointVO class* | *The watchlist is being returned to a list of type DataPointsVO, so navigated to that class.* |
| 7 | *Next, we moved on to the MangoValue class. In the MangoValue class, we spotted a objectToValue method.* | *Spotted a mangoValue object and identified that it is the origin of values, so navigated there.* |
| 8 | *Next, went to the NumericValue class. This class has a instance variable "value".* | *We were not sure if this method had to be changed, therefore we changed the value there and checked, it was impacting all the data.* |
| 9 | *Therefore, we spotted that the value attribute in the NumericValues constructor has to be changed.* | *We confirmed this attribute had to be modified.* |

**Time spent (in minutes):** 120

*Classes and methods inspected:*

*mangoSource/src/com/serotonin/mango/web/dwr/WatchListDwr.java*
*public WatchListState addToWatchList(int pointId)*

*mangoSource/src/com/serotonin/mango/db/dao/WatchListDao.java*
*public void populateWatchlistData(WatchList watchList)*

*mangoSource/src/com/serotonin/mango/vo/DataPointVO.java*
*public void resetLastValue()*
*mangoSource/src/com/serotonin/mango/rt/dataImage/types/MangoValue.java*
*public static MangoValue stringToValue(String valueStr, int dataType)*

*mangoSource/src/com/serotonin/mango/rt/dataImage/types/NumericValue.java*
*public NumericValue(double value)*

## 2. Impact Analysis

| Step # | Description | Rationale |
|---|---|---|
| *1* | *Run the system.* | *To ensure the system is operational and ready for interaction or testing.* |
| *2* | *Interact with the system by logging in and accessing the schedule screen.* | *To familiarize oneself with system features and identify areas for potential modification or enhancement.* |
| *3* | *Perform a workspace search using the "watchlist" keyword via Ctrl+H.* | *To gather additional insights into the implementation and usage of the watchlist feature within the system.* |
| *4* | *Navigate to the WatchListDwr.java file and understand the data retrieval process.* | *To comprehend how data is fetched within the system and understand dependencies on other classes.* |
| *5* | *Identify and explore the populateWatchListData method within the WatchListDao class.* | *To determine the method responsible for populating watchlist data and understand data flow within the system.* |
| *6* | *Investigate the DataPointVO class due to its relevance to the watchlist feature.* | *To understand the structure and behavior of components associated with the watchlist feature.* |
| *7* | *Examine the MangoValue class and locate the objectToValue method.* | *To identify the method responsible for generating values and understand data processing mechanisms.* |
| *8* | *Explore the NumericValue class and observe the presence of the "value" instance variable.* | *To assess the impact of modifying the "value" attribute on data integrity and system functionality.* |
| *9* | *Confirm the necessity of modifying the value attribute in the NumericValues constructor.* | *To ensure that modifications align with identified requirements and objectives based on observed data changes.* |

**Time spent (in minutes):** 40 minutes

*mangoSource/src/com/serotonin/mango/web/dwr/WatchListDwr.java*
*public WatchListState addToWatchList(int pointId)*

*mangoSource/src/com/serotonin/mango/db/dao/WatchListDao.java*
*public void populateWatchlistData(WatchList watchList)*

*mangoSource/src/com/serotonin/mango/vo/DataPointVO.java*
*public void resetLastValue()*

*mangoSource/src/com/serotonin/mango/rt/dataImage/types/MangoValue.java*
*public static MangoValue stringToValue(String valueStr, int dataType)*

*mangoSource/src/com/serotonin/mango/rt/dataImage/types/NumericValue.java*
*public NumericValue(double value)*

## 3. Pre-factoring

| ***Step #*** | ***Description*** | ***Rationale*** |
|---|---|---|
| ***1*** | *We performed initial inspection in the files to check if Prefactoring can be performed.* | |
| ***3*** | *No explicit Pre-factoring was performed. We made no changes with git.* | *Since a single line of code from which the value was populated change would meet the requirement to make the desired change, there was no necessity found to perform Prefactoring.* |

## 4. Actualization

| *Step #* | *Description* | *Rationale* |
|---|---|---|
| *1* | *We initiated the system to ensure its operational status.* | *Verifying the operational status ensures that subsequent steps are conducted in an environment ready for exploration.* |
| *2* | *Following authentication, we accessed the schedule screen to explore system functionalities...* | *Exploring system functionalities aids in identifying areas for potential improvement or modification.* |
| *3* | *Utilizing the Ctrl+H shortcut, we conducted a workspace search using the "watchlist" keyword...* | *Conducting a keyword search provides deeper insights into the implementation and usage of specific system features.* |
| ***4*** | *We navigated to the WatchListDwr.java file to comprehend the data retrieval process...* | *Understanding data retrieval processes helps in determining the sources and mechanisms underlying system functionality.* |
| *5* | *Within the WatchListDao class, we identified the populateWatchListData method...* | *Identifying methods responsible for data population provides clarity on data flow and manipulation within the system.* |
| *6* | *Proceeding to the DataPointVO class, we investigated its relevance to the watchlist feature...* | *Investigating relevant classes aids in understanding the structure and behavior of components associated with specific features.* |
| *7* | *Next, we examined the MangoValue class, where we located the objectToValue method...* | *Identifying methods responsible for value generation facilitates understanding of data processing and transformation.* |
| *8* | *Advancing to the NumericValue class, we observed the presence of the "value" instance variable...* | *Observing variable usage and behavior assists in assessing potential impacts on data integrity and system functionality.* |
| *9* | *Consequently, we confirmed the necessity of modifying the value attribute in the NumericValues constructor...* | *Confirming the need for modifications ensures alignment with identified requirements and objectives.* |

**Time spent (in minutes):** 40 minutes

*mangoSource/src/com/serotonin/mango/web/dwr/WatchListDwr.java*
*public WatchListState addToWatchList(int pointId)*

*mangoSource/src/com/serotonin/mango/db/dao/WatchListDao.java*
*public void populateWatchlistData(WatchList watchList)*

*mangoSource/src/com/serotonin/mango/vo/DataPointVO.java*
*public void resetLastValue()*

*mangoSource/src/com/serotonin/mango/rt/dataImage/types/MangoValue.java*
*public static MangoValue stringToValue(String valueStr, int dataType)*

*mangoSource/src/com/serotonin/mango/rt/dataImage/types/NumericValue.java*
*public NumericValue(double value)*

## 5. Validation

| Step # | Description | Rationale |
|---|---|---|
| **1** | *Test case defined:*<br>*Inputs: logging in into the website*<br>*Expected output : the values in the watchlist should be truncated to 2 decimal points* | *This is the regular expected behavior. The test passed.* |
| **2** | *Test case defined:*<br>*Inputs: clicked on the point details icon*<br>*Expected output: the values in the temp_table, statistics table and the history table should be truncated to 2 decimal points* | *This is the regular expected behavior. The test passed.* |
| **3** | *Test case defined:*<br>*Inputs: There is a "point" section on the left side of the website, there is an arrow which says "Add to watch list". Click on it.*<br>*Expected output : A new row is populated. The values in that row should also be truncated to 2 decimal points.* | *This is a non-trivial test case.*<br>*This test passed.* |
| | | |

**Time spent (in minutes):** 40

## 6. 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 | 120 | 5 | 1 | 5 | 1 |
| Impact Analysis | 40 | 5 | 1 | 5 | 1 |
| Prefactoring | | | | | |
| Actualization | 40 | 5 | 1 | 5 | 1 |
| Postfactoring | 0 | 0 | 0 | 0 | 0 |
| Verification | 40 | | | | |
| **Total** | 240 | | | | |

## 7. Conclusions

*Experience with Change Request:*
*The change request was straightforward and, but the concept location part was a bit tricky.*
*The search functionality and debugging tools were a boon.*
*The testing was simple and covered all necessary scenarios.*