# CS/SE 6356 Software Maintenance, Evolution & Re-engineering
## Spring 2024
## Assignment 1: Mango changes

Due date: 02/21/2024 (end of day)

**This is a team assignment.  You are allowed to exchange information with other teams about installation, but NOT about the changes.**

# 1   Goals of the assignment

In this assignment, you are to implement three change requests for Mango, a web application written in Java.

The source code and the instructions for installing Mango are available in the following OneDrive folder:

https://cometmail.sharepoint.com/:f:/s/CS6356001SE6356001SYSM63080012242/EoKlIhUTXshKgwcNRPkQ6TkBUApApywFOv2a54HfY5kb8Q?e=O0W9YD

Those instructions include the required software for Mango deployment and the installation guide on both Linux and Windows systems.  (Note: If you are using MacOS, please refer to the guide of installation on Linux.)

**List of required software:**

- Apache Ant 1.10.x - Java automated build + deployment tool
- Apache tomcat 9.0.x - Web Server
- Java 8 JDK - Java
- mangoSource - Mango source code
- mangoDB - Mango database

Read carefully the assignment before proceeding.

# 2   Change requests

Each team will have to make all three of the following change requests and implement them in Mango.

## 2.1   Change request #1

Mango has an interface called the watch list that allows a user to view the stored data of a set of sensors. Alternatively, each sensor can be viewed independently of one another with the "point details" interface. Both of these tools display the collected values with unnecessary precision. (see **Error! Reference source not found.**).



*Figure 1.  Two Decimal Truncation in Mango – Current Behavior*

You are requested to make changes so that Mango should display all values with at most two decimal places of precision. These values should be truncated, not rounded. This change should be made anywhere that values are displayed. (see Figure 2.)
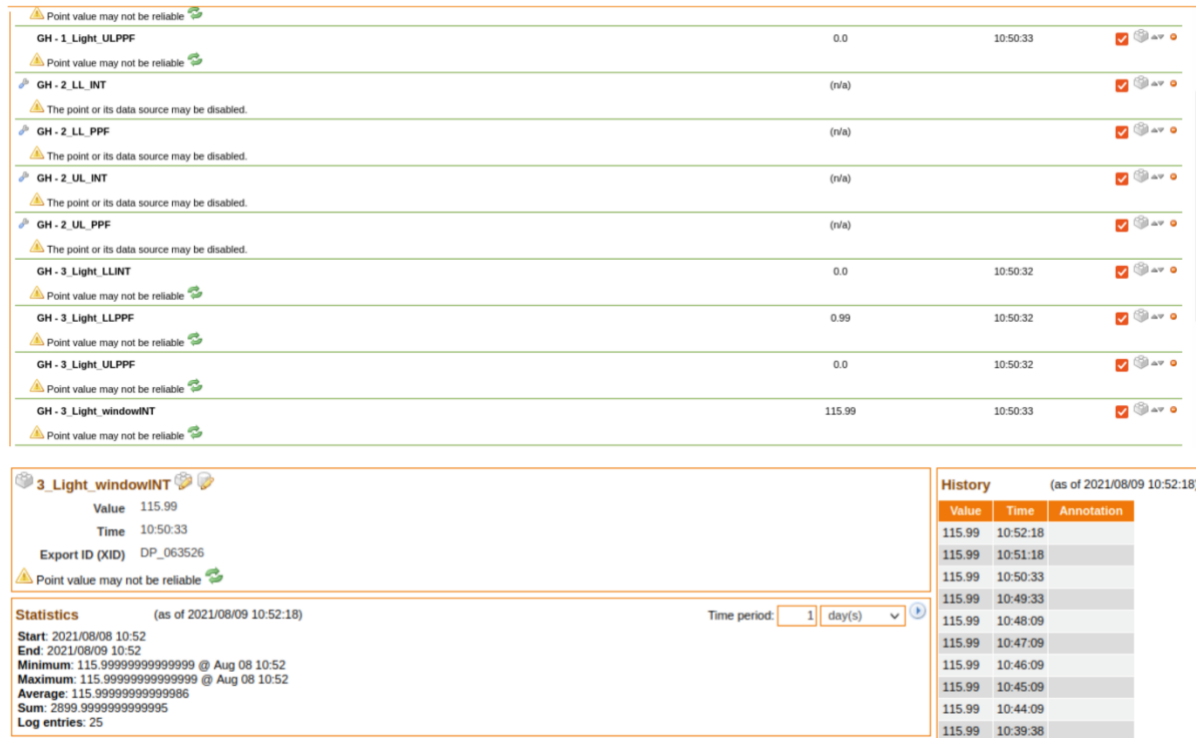
*Figure 2. Two Decimal Truncation in Mango – Expected Behavior*

## 2.2 Change request #2

By default, a notification sound plays by default when certain events occur, including when a sensor records a new value. This feature can be toggled by clicking the mute button on the right side of the navigation bar. (see Figure 3.)
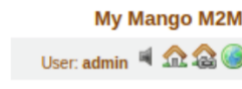


*Figure 3. Muted by Default in Mango – Current Behavior*

This feature may become useless and annoying if a large amount of sensors are constantly recording information, so it should be disabled by default. It should not be removed completely, and it should still be possible to enable and disable the notification sound. (see Figure 4.)
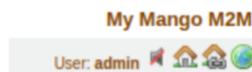


*Figure 4. Muted by Default in Mango – Expected Behavior*

## 2.3 Change request #3

When attempting to change an administrator's password in the "Users" page, an error occurs and the password is not changed. (see Figure 5.)
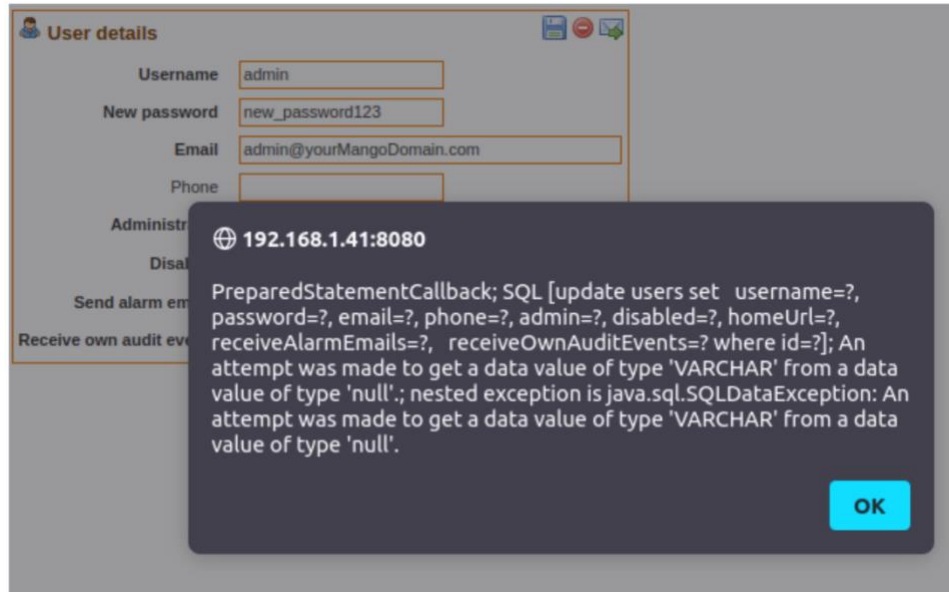


*Figure 5. Change Password Bug in Mango – Current Behavior*

This error should be fixed, allowing an administrator's password to be changed without an error popup appearing. (see Figure 6.)
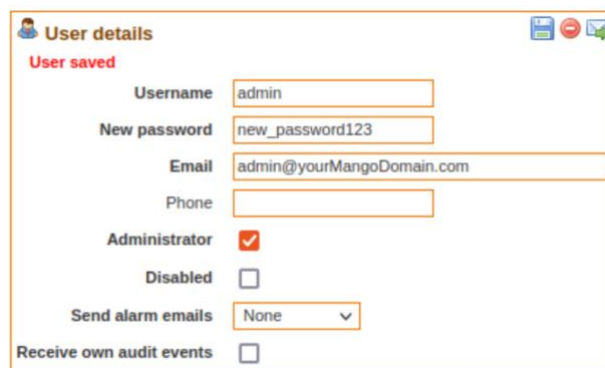


*Figure 6. Change Password Bug in Mango – Expected Behavior*

# 3   What are you supposed to do?

1. Create a private GitHub repository called MangoTeamXX, where XX is your team number. If you do not have a GitHub account, please get one. Share the repository with with ***mohPYdev*** and ***ChaoticCross12***.

2. Copy the source code from the shared OneDrive folder. The mangoDB.zip file contains the Mango database and the mangoSourceJava8.zip file contains Java source code. You will need both to compile, install, and run Mango.
3. Follow the instructions from the Mango Installation Guide.pdf and make sure you can compile and run the software.
4. Run the system to familiarize yourself with its functionality.
5. Select the changes you will do for this assignment.

For each change request, complete the following steps:

6. Read and understand the change request. Ask any question you may have about the change requests to the TA. Last-day questions are strongly discouraged and will not help you finish the assignment on time.
7. Create a branch in the repository called "mango-cr-<#>", where <#> is the change request number. For example, for change request #3, the branch should be called "mango-cr-3". After you make your change, with possible multiple commits in the branch, merge the change branch into the main branch. This way each change can be analyzed independently.
8. You will have to work together. Follow the change process covered in class (i.e., concept location, impact analysis, actualization, etc.) to implement the change request. Prefactoring and postfactoring are optional. During the change, one team member will change the code in the IDE while the other will track the activities and record the major decisions taken by the team using the change request log (see the log template for specific instructions). Feel free to decide who changes the code and who tracks the process. Ideally, both team members should alternate their roles for each change request. In order for this to work, you will have to discuss out loud each decision. You can do this in person or via MSTeams or other virtual collaboration tool. Do not work asynchronously, meaning that one student makes the changes and then later the other (or the same one) completes the log. This will result in incomplete logs and defeats the purpose of the later assignments, where you will analyze the logs produced by all teams.
9. After you make your change, perform functional testing by executing several scenarios related to the features affected by the change. Make sure you define a set of test cases (i.e., usage scenarios) and keep track of the ones that fail and pass. Track the testing activity in the change request log. If you want to add unit test, feel free to do so, but it is optional.
10. After you implement and test the change request, push your local changes to the remote GitHub repository. Feel free to make several commits to your local working copy before pushing.
11. Name the log file as follows: mango-cr#X-teamYY.pdf, where X is the number of the change request and YY is the number of your team. You may submit the log as docx or pdf.

# 4 Deliverables

There are three types of deliverables for this assignment:

1. **Modified source code of the system in GitHub**.
   a. Make sure you have one separate branch for each change request
   b. Remember to add the TAs as collaborator of your repositories (GitHub accounts: *mohPYdev* and *ChaoticCross12*)

2. **Change log for each change request**. Consider the next instructions:

   a. Fill-in the change log form for each implemented change request while implementing the change requests.

   b. Add the log forms to your GitHub repository.

   c. Submit the logs through eLearning (one log file per change request)

3. Contribution information. Add another file where you include the name of each team member and how much they contributed to this assignment (in percentages). For example: John Doe 40%; Jane Doe 60%. The percentage should add up to 100%. Note. Read the information on eLearning about how the assignments will be graded.

# 5 Notes

- The logs should not contain any gap in the process you followed. Provide enough details about queries, tools used, classes visited, decisions, etc. Any person should be able to precisely understand and replicate the process you followed. It is very important to clearly describe the rationale of your tasks and decisions.

- Do not forget to evaluate the process you went through and provide enough details in the conclusions in the change request logs. Include details on why some steps were harder than others.

- Proofread carefully your logs.

# 6 Grading

| Item | Points |
|------|--------|
| Repository and branch creation | 10 |
| Code changes, correct code compilation, and correct change requests implementation | 50 |
| Code change description (in the change log) | 40 |
| **Total** | **100** |

**Notes:** Proofread carefully your scenario descriptions and change request logs. You will receive points off in case of writing mistakes, gaps in the logs, compilation errors.