Change request log

1. Concept Location

Step #	Description	Rationale
1	We successfully installed and deployed the necessary tools and software.	To setup the environment and ensure that the application runs without any issues.
2	We ran the Mango application and logged in.	To understand the functionality of the application, and to locate the mute button.
3	We explored the Mango application's core features and functionality by going through its various pages and modules.	To confirm the application's expected behavior.
4	We performed a search across the source code to find the files related to the mute button. Using the command-line tool 'grep -R' in the terminal, we searched for the string "mute" to identify potential sources of the mute functionality.	To find the relevant code responsible for the mute functionality. We expected to find related code in event handlers.
5	We found the 'User.java' file and found the' private transient Boolean muted = false;'. This line directly affects the default state of the "muted" setting for a user.	This code line in the 'User.java' file is important because it is used to enable the notification sound by default.
6	We navigated to the 'page.tag' file under '/mangoSource/war/WEB-INF/tags/page.tag'. This file contains the navigation bar.	Exploring this file seemed relevant since the mute button is on the navigation bar. We confirmed this file needs to be modified.
7	We reviewed 'page.tag' and found 'setUserMuted(${sessionUser.muted});'. This line was responsible for initiating the mute state based on the user's session.	Exploring this file seemed important because the mentioned code line in the description directly initializes the mute state. We confirmed this code line needs to be modified.
8	We then looked at the 'header.js' file, which contains the 'SoundPlayer()' function that controls sound playback and muting.	Exploring this file also seemed important because it is responsible for the sound player. We confirmed this function needs to be modified.
9	We reviewed the 'common.js' file, especially the 'mango.longPoll.pollCB' function. We found that this function is responsible for generating alerts, but it also plays a sound whenever a new event happens.	We confirmed this function needs to be modified since it is responsible for triggering the sounds.
10	We also reviewed files such as 'soundmanager2-nodebug-jsmin.js' and 'soundmanager2.js' since they contained information related to the second change request.	We confirmed that these files do not need to be modified because they are the core library for sound management.

Time spent (in minutes): 150 minutes

Classes and methods inspected:

- /mangoSource/src/com/serotonin/mango/vo/User.java
  - isMuted()
  - setMuted()
- /mangoSource/war/WEB-INF/tags/page.tag
  - setUserMuted(${sessionUser.muted});
- /mangoSource/war/resources/header.js'
  - SoundPlayer()
- /mangoSource/war/resources/common.js
  - mango.longPoll.pollCB()
- /mangoSource/war/resources/soundmanager2-nodebug-jsmin.js

    - SoundManager()
- /mangoSource/war/resources/soundmanager2.js
    - SoundManager()

2. Impact Analysis

Step #	Description	Rationale
1	We determined that the 'User.java' file is where the default value of mute should be changed to true. This will cause the sound to be muted when the new user sessions begin, which affects the user experience.	The muted state needs to be changed for the session to start with a disabled notification sound. We realized that the mute state is the only item that needs to be changed in the 'User.java' file. The 'isMuted()' and 'setMuted()' methods don't need to be changed.
2	The 'isMuted()' and 'setMuted()' methods in the 'User.java' don't need any modification.	We determined that these methods don't need any changes because both methods correctly handle reading and setting the mute state.
3	We determined that 'page.tag' is an important file. Updating 'setUserMuted(true)' in 'dojo.addOnLoad(function(){setUserMuted(true);});' to set the value to be muted by default is required to change the status by default.	The 'page.tag' needed to be changed because we are changing the original mute status. The main impact of this change is that the user will be muted by default. The user's audio will be muted when they first load any page that uses this 'page.tag.'
4	We determined that we need to change the mute property's state within the 'SoundPlayer()' function in the 'header.js' file. This will make the default state of the sound player to be muted.	This change directly affects the initialization of the SoundPlayer object, which is used throughout the application to handle audio notifications.
5	To handle the mute/unmute functionality, we found the 'common.js' file. This file handles whether to play the sounds or not.	This change impacts the playing of the sounds, which is a necessary change for this change request.

Time spent (in minutes): 30 minutes

Classes and methods inspected:

- /mangoSource/src/com/serotonin/mango/vo/User.java
    - isMuted()
    - setMuted()
- /mangoSource/war/WEB-INF/tags/page.tag
    - setUserMuted(${sessionUser.muted});
- /mangoSource/war/resources/header.js'
    - SoundPlayer()
- /mangoSource/war/resources/common.js
    - mango.longPoll.pollCB()

3. Prefactoring (optional)

Step #	Description	Rationale
1	N/A	N/A
2	N/A	N/A
3	N/A	N/A

4	N/A	N/A

Time spent (in minutes): N/A

4. Actualization

Step #	Description	Rationale
1	Changed 'private transient boolean muted = false' to 'private transient boolen muted = true' in the User.java file, which is located in the /mangoSource/src/com/serotonin/mango/vo/User.java.	We realized that making this change will set the default mute state to true for new user sessions.
2	Changed 'dojo.addOnLoad(function(){setUserMuted($sessionUser.muted});' to 'dojo.addOnLoad(function(){setUserMuted(true);});' in the page.tag file, which is located in the /mangoSource/war/WEB-INF/tags/page.tag.	We made this change to set the default mute state to true.
3	Changed the 'this.mute = false' to 'this.mute = true' in the SoundPlayer() method located in the header.js file. The header.js file is located in /mangoSource/war/resources/header.js.	To make sure that the sound is muted by default.
4	Added '&& !mango.soundPlayer.muted' for mango.longPoll.CB in the common.js file, which is located in the /mangoSource/war/resources/common.js.	To make sure the sounds do not play on default.
5	Rebuilt the Mango application using Ant.	To apply the code changes.

Time spent (in minutes): 40 minutes

Classes and methods inspected:

- /mangoSource/src/com/serotonin/mango/vo/User.java
- /mangoSource/war/WEB-INF/tags/page.tag
  - setUserMuted(${sessionUser.muted});
- /mangoSource/war/resources/header.js'
  - SoundPlayer()
- /mangoSource/war/resources/common.js
  - mango.longPoll.pollCB()

5. Postfactoring (optional)

Step #	Description	Rationale
1	N/A	N/A
2	N/A	N/A
3	N/A	N/A
4	N/A	N/A

Time spent (in minutes): N/A

6. Validation

Step #	Description	Rationale

1	Test case defined: Default mute on application start. Inputs: Mango application successfully reinstalled and reloaded. Expected output: Notification sound should be muted by default.	This is the expected behavior. The test passed. The notification sound is muted by default.
2	Test case defined: User events should not trigger sound. Inputs: User logged in and triggers an event. Expected output: No sound should play.	This is the expected behavior. The test passed. The sound was muted, the UI showed muted status, and no sounds played.
3	Test case defined: Sound toggle functions correctly. Inputs: User logged in and clicks the sound button to unmute. Expected output: Notification sound should be enabled.	This is the expected behavior. The test passed. The notification sound is enabled, and the UI reflects this change.
4	Test case defined: Application defaults to muted even after browser reload (without rebuild). Inputs: User reloads the web page. Expected output: Notification sound should be muted by default.	This is the expected behavior. The test passed. Notification sound remains muted by default, even after enabling the button before page reload.

Time spent (in minutes): 25 minutes

Classes and methods inspected:

- /mangoSource/src/com/serotonin/mango/vo/User.java
- /mangoSource/war/WEB-INF/tags/page.tag
  - setUserMuted(${sessionUser.muted});
- /mangoSource/war/resources/header.js'
  - SoundPlayer()
- /mangoSource/war/resources/common.js
  - mango.longPoll.pollCB()

7. 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	150	6	0	7	0
Impact Analysis	30	4	0	5	0
Prefactoring	0	N/A	N/A	N/A	N/A
Actualization	40	4	4	3	3
Postfactoring	0	N/A	N/A	N/A	N/A
Verification	25	4	0	3	0
Total	245	18	4	18	3

8. Conclusions

This change request was relatively simple to implement due to the well-defined instructions and clear objectives. The focus of this change request was to ensure that the notification sound defaulted to mute, which only required minor code adjustments across several files. To efficiently locate the relevant files, especially those controlling the mute button, we used the 'grep – R' command, which was highly effective.

However, we encountered several issues during the initial installation and environment setup, which made it a bit challenging. Along with the initial setup, grasping the code structure also proved to be a challenge. While the code change itself was simple, understanding how the notification system worked within the application required thorough investigation. The biggest challenge, however, was verifying that the code change behaved as expected. Since multiple files were modified, we needed to make sure that the mute setting was consistently applied throughout the whole application. To address this, we conducted extensive testing after each modification.

Overall, this change request process showed us how minor code modifications can have a huge impact on the user experience. This change request reinforced the importance of a strong understanding of the application's structure before making any modifications. We also understood the value of robust testing to ensure that minor code adjustments work correctly in all scenarios. Through this process, we the importance of attention to detail, problem-solving, and persistence when working with existing codebase.