# Change Request Log

# Team

Team 2: Michelle Jones and Rangy Samuel

# Change Request

FEMR-158 <https://teamfemr.atlassian.net/browse/FEMR-158>

# Concept Location

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | We ran the system | Can’t fix what we can’t run. |
| 2 | We interacted with the system: after logging in we went to the admin panel then selected the user’s tab. We choose the first user on the list to edit. | Change request lists bug occuring when admin attempts to edit a user’s profile. Best place to start is there… |
| 3 | Recreated the bug by applying no changes to see the error message. | Had to verify the bug to understand the issue.  Error text in red when no input is made to either password field is “password field is empty” |
| 4 | Open cmd prompt and performed grep search on femr project for the string “Change User Password\*\*” | Unique string that would point directly to the form that is called. Work our way from there. |
| 5 | Examined file “edit.scala.html” | File contained “Change User Password\*\*” string. Further examination showed it was responsible for showing the user the form in html. |

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

# Impact Analysis

Use the table below to describe each step you followed when performing impact analysis for this change request. Include as many details as possible, including why classes are visited or why they are discarded from the ones that have to change.

Do not take the impact analysis of your changes lightly. Remember that any small change in the code could lead to large changes in the behavior of the system. Follow the process on impact analysis covered in the class. Describe in details how you followed this process in the change request log. Provide details on how and why you finished the impact analysis process.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | Examined file “UsersController.java” | Edit.scala.html lists the file as an import. The line where the form adds the user password fields lists a @helper. The helper class appears to be using this class…. |
| 2 | Examined “EditModelView.java” | Edit.scala.html lists that this file is used as the form! Figured there has to be logic in there for the password entry. |
| 3 | EditModelView.java:Line 52 – Found the error message “password field is empty”. | It appears that the if/else if/else statement is throwing an error if both fields are empty.  This if/else if/else statement is contained within a validate() function. Going to see where that is called next, by going and seeing what happens when you click the “Save” button in Edit.scala.html |
| 4 | Re-examined “Edit.scala.html” | The save button submits the form, but doesn’t specifically call a validate function. But, thanks to previous knowledge of html and forms, I know that the form has to specify an action to take when submitted. Going to look at UsersController.java, since that is what the form is based on |

**Time spent (in minutes):** 15

# Actualization

Using the table below, describe each step you followed when changing the code. Include as many details as possible, including why classes/methods were modified, added, removed, renamed, etc.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | Added an if statement within the evaluation specifically for password length being tested on both the new password and the new password verify fields. | The if/else if/else statements tested to see if either fields were empty to throw an error, but did not take into account that if both fields were empty (&& instead of ||), that the admin hadn’t modified the password field at all. Therefore there shouldn’t be an error if both fields are empty. |
| 2 | Ran the system in debug mode with a break point at the else if that contains the newly added if statement | Wanted to make sure that the if would be examined during a run and see the flow of the program if nothing was entered in the fields. |

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

# Validation

Using the table below, describe any validation activity (e.g., testing, code inspections, etc.) you performed for this change request. Include the description of each test case, the result (pass/fail) and its rationale.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | Rebuilt using activator clean install | Wanted to see if any tests would fail. This would point me to the file(s) that would need updating (if any).. |
| 2 | No errors occurred during rebuild | Appears that a test doesn’t need to be modified, but that doesn’t mean that one shouldn’t be created. |

**Time spent (in minutes):** 5

# Timing

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 10 |
| Impact Analysis | 15 |
| Prefactoring | 0 |
| Actualization | 10 |
| Postfactoring | 0 |
| Verification | 5 |
| Total | 40 |

# Reverse engineering

# 

# Conclusions

For this change, it was relatively simple to find the location of the bug, since it was occuring from the UI. Locating the logic from that as a starting point greatly helped. What isn't taken into account for this change request is initial environment setup, which took hours to complete. I originally started using Windows and was unable to get it to work, so I ended up making my laptop dual boot Windows 10 and Ubuntu just so I could perform the work on a Linux machine. That took around 8 hours to set up.

Classes and methods changed:

* [femr](https://github.com/fearlesMICHELLE/femr/tree/FEMR-158)/[app](https://github.com/fearlesMICHELLE/femr/tree/FEMR-158/app)/[femr](https://github.com/fearlesMICHELLE/femr/tree/FEMR-158/app/femr)/[ui](https://github.com/fearlesMICHELLE/femr/tree/FEMR-158/app/femr/ui)/[models](https://github.com/fearlesMICHELLE/femr/tree/FEMR-158/app/femr/ui/models)/[admin](https://github.com/fearlesMICHELLE/femr/tree/FEMR-158/app/femr/ui/models/admin)/[users](https://github.com/fearlesMICHELLE/femr/tree/FEMR-158/app/femr/ui/models/admin/users)/**EditViewModel.java**