Change request log

1. Concept Location

Step #	Description	Rationale
1	We ran the system	To establish a baseline understanding of system behavior.
2	We interacted with the system: after logging in we clicked on different icons on the navigation bar and identified the user page.	To get familiar with system features and locate the user page where password changes and error popups occur.
3	From the list of users available, we clicked on the admin user and tried changing the password	To replicate the issue and observe the error popup
4	Before pressing the save button, we right clicked → selected inspect →went to network tab →Collected the list of objects visible under the Name column UsersDwr.saveUserAdmin.dwr	To capture relevant requests, responses for analysis and get a list of keywords which can be useful to perform grep
5	We clicked on the 1st item to explore the details under Payload tab and Response tab	To identify the exact data being sent and received.
6	Conducted a global search in the IDE for "saveUserAdmin".	To locate the code responsible for handling password updates.
7	Found 9 results, narrowed down to Mango.log, users_jsp.java, and UsersDwr.java.	To focus only on the most relevant files and methods.
8	Ranked the concepts by the likelihood that they can be easily located, we first checked the Mango.log and again identified UsersDwr.java class and the saveUserAdmin method. Further another new class UserDao.java was found	To backtrack and identify other potential classes and source of error
9	Since we identified UsersDwr the third time we decided to inspect it further. We accessed UsersDwr.java using the IDE file locator	To inspect how the method interacts with the database.
10	Inspected saveUserAdmin method, verifying parameter list similarity to network request.	To confirm the data flow aligns with the observed issue.
11	We found 3 methods of UserDao class being used and decided to navigate to the saveUser method under UserDao using dependency navigator of the IDE	To trace how data is being stored or updated in the database.
12	We found two methods here: insertUser and updateUser. We decided to explore updateUser using dependency navigator of IDE	To determine the exact point where the error originates we inspected methods responsible for updating passswords.
14	We noticed an update query being run on users table. This query was like the query seen in the error popup	To confirm if the update query logic needs adjustment.
15	We tired adding a breakpoint on line 148 and ran the program in debugging mode.	To track the execution flow and pinpoint issues dynamically.
16	We faced problems in triggering the breakpoint because of some standard debugger error, so moved forward with other methods for concept location like - Concept location via dependency search and dynamic search	To ensure concept location despite debugger constraints.
17	We decided to study the source of the data being populated in the updateMethod and identified the getter and setter methods in the User.java class	To validate how user attributes are being managed.
18	We noticed that the users_jsp.java class line 856 in method writing content to the JSP response output	To determine if UI-level changes are necessary.
18	We used grep to search for terms like “users” and “.sql” and found the same schema in multiple “.sql”	To verify database integrity and avoid unintended changes.

	files and noted down the type and constraints in the users table	
19	We then checked different files where this table was being updated and again landed up on the same point as before	To finalize which class needs modification.
20	After re studying the log, considering the schema and searching for update queries on users table we marked the UserDao.java as "located".	We confirmed this class had to be modified.

Time spent (in minutes): 190

Classes and methods inspected:

- */Users/sakshitokekar72/Documents/Spring2025/SoftwareMaintenance/Assignment1/MangoTeam02/Documents/mango/mangoSource/src/com/serotonin/mango/db/dao/UserDao.java*
  - *public User getUser(int id)*
  - *public List<User> getUsers()*
  - *public void saveUser(final User user)*
  - *void updateUser(User user)*

- */Users/sakshitokekar72/Documents/Spring2025/SoftwareMaintenance/Assignment1/MangoTeam02/Documents/mango/mangoSource/src/com/serotonin/mango/vo/User.java*
  - *public void validate(DwrResponseI18n response)*
  - *public String getPhone()*
  - *public void setPhone(String phone)*
  - *public int getId()*
  - *public void setId(int id)*
  - *public String getPassword()*
  - *public void setPassword(String password)*
  - *public String getUsername()*
  - *public void setUsername(String username)*
  - *public boolean isAdmin()*
  - *public void setAdmin(boolean admin)*
  - *public String getEmail()*
  - *public void setEmail(String email)*
  - *public boolean isDisabled()*
  - *public void setDisabled(boolean disabled)*
  - *public String getHomeUrl()*
  - *public void setHomeUrl(String homeUrl)*
  - *public int getReceiveAlarmEmails()*
  - *public void setReceiveAlarmEmails(int receiveAlarmEmails)*
  - *public boolean isReceiveOwnAuditEvents()*
  - *public void setReceiveOwnAuditEvents(boolean receiveOwnAuditEvents)*

- */Users/sakshitokekar72/Documents/Spring2025/SoftwareMaintenance/Assignment1/MangoTeam02/Documents/mango/apache-tomcat-9.0.98/work/Catalina/localhost/test/org/apache/jsp/WEB_002dINF/jsp/users_jsp.java*
  - *public void invoke0(javax.servlet.jsp.JspWriter out)*

- */Users/sakshitokekar72/Documents/Spring2025/SoftwareMaintenance/Assignment1/MangoTeam02/Documents/mango/mangoSource/src/com/serotonin/mango/web/dwr/UsersDwr.java*
  - *public DwrResponseI18n saveUserAdmin(int id, String username, String password, String email, String phone, boolean admin, boolean disabled, int receiveAlarmEmails, boolean receiveOwnAuditEvents, List<Integer> dataSourcePermissions, List<DataPointAccess> dataPointPermissions)*
  - *public DwrResponseI18n saveUser(int id, String password, String email, String phone, int receiveAlarmEmails, boolean receiveOwnAuditEvents)*

2. Impact Analysis

Step #	Description	Rationale
1	Classes identified in concept location made up the initial impact set. We made a list of files, classes and methods that call the updateUser method. This method had local functionality	To track all components potentially affected by the change.
2	updateUser method of class UserDao.java needs to be changed. We need to check for null values before updating the table. Class dependencies was analyzed, and status of components were updated	Because it is responsible for updating values in the users table and this is where we will make the changes that causes the error to popup
3	The class UsersDwr.java was discarded from the list of classes to change and marked as unchanged as on inspection the class it was found that it was not impacted by the change.	Because the method only deals with making changes to values in database and doesn't return anything
4	The class Users_jsp.java was discarded from the list of classes to change and marked as unchanged as on inspection the class it was found that it was not impacted by the change	Because the method only deals with making changes to values in database and doesn't return anything

Time spent (in minutes): 45

- */Users/sakshitokekar72/Documents/Spring2025/SoftwareMaintenance/Assignment1/MangoTeam02/Documents/mango/mangoSource/src/com/serotonin/mango/db/dao/UserDao.java*
    - *public User getUser(int id)*
    - *public List<User> getUsers()*
    - *public void saveUser(final User user)*
    - *void updateUser(User user)*

- */Users/sakshitokekar72/Documents/Spring2025/SoftwareMaintenance/Assignment1/MangoTeam02/Documents/mango/mangoSource/src/com/serotonin/mango/web/dwr/UsersDwr.java*
    - *public DwrResponseI18n saveUserAdmin(int id, String username, String password, String email, String phone, boolean admin, boolean disabled, int receiveAlarmEmails, boolean receiveOwnAuditEvents, List<Integer> dataSourcePermissions, List<DataPointAccess> dataPointPermissions)*
    - *public DwrResponseI18n saveUser(int id, String password, String email, String phone, int receiveAlarmEmails, boolean receiveOwnAuditEvents)*

- */Users/sakshitokekar72/Documents/Spring2025/SoftwareMaintenance/Assignment1/MangoTeam02/Documents/mango/apache-tomcat-9.0.98/work/Catalina/localhost/test/org/apache/jsp/WEB_002dINF/jsp/users_jsp.java*
    - *public void invoke0(javax.servlet.jsp.JspWriter out)*

3. Prefactoring **(optional)**

Prefactoring was not needed

Step #	Description	Rationale
1	...	

Time spent (in minutes): 0

4. Actualization

Step #	Description	Rationale
1	We created a validation function checkHomeUrl	We realized that the homeUrl value was being set null and it was causing the error popup. That's why we created a function which checks for HomeUrl value.
2	Before updating the values in the users table we check if homeUrl is null and replace it with *http://localhost:8080/test/watch_list.shtm* which is the home page URL	When we did a grep global search for homeUrl, we noticed in the */Users/sakshitokekar72/Documents/Spring2025/SoftwareMaintenance/Assignment1/MangoTeam02/Documents/mango/mangoSource/war/WEB-INF/classes/messages_en.properties* file that homeUrl is the home page URL
3	We maintained a change log at the top of the class describing the change details	For easier tracking and understanding of the change request whenever required in the future.
4	We also ran the existing test cases.	To make sure everything works.

Time spent (in minutes): 75

Classes and methods inspected:

- */Users/sakshitokekar72/Documents/Spring2025/SoftwareMaintenance/Assignment1/MangoTeam02/Documents/mango/mangoSource/src/com/serotonin/mango/db/dao/UserDao.java*
    - *public User getUser(int id)*
    - *public List<User> getUsers()*
    - *public void saveUser(final User user)*
    - *void updateUser(User user)*

Classes and methods changed:

- */Users/sakshitokekar72/Documents/Spring2025/SoftwareMaintenance/Assignment1/MangoTeam02/Documents/mango/mangoSource/src/com/serotonin/mango/db/dao/UserDao.java*
    - *void updateUser(User user)*
    - *private String checkHomeUrl(String value)*

5. Postfactoring (optional)

Postfactoring was not needed

Step #	Description	Rationale
1	...	

Time spent (in minutes): 0

6. Validation

Step #	Description	Rationale
1	Test case defined: Inputs: abcd1234 Expected output: successful re login with new password	This is the regular expected behavior. The test passed.
2	Test case defined: Inputs: qwerty Expected output: successful re login with new password	This is the regular expected behavior. The test passed.

Time spent (in minutes): 40

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	190	4	0	27	0
Impact Analysis	45	3	0	7	0
Prefactoring	0	0	0	0	0
Actualization	75	2	1	5	1
Postfactoring	0	0	0	0	0
Verification	40	0	0	0	0
Total	350	9	1	39	1

8. Conclusions

This change request was executed efficiently due to the structured approach to concept location, impact analysis, and actualization. The key challenges included:

- Identifying the root cause of the homeUrl issue.
- Coming up with the best possible solution
- Debugging difficulties, which required alternate search strategies.
- Ensuring that modifications did not introduce unintended side effects.

The process highlighted the importance of systematic analysis and validation through test cases to confirm a successful resolution.