

Three bugs:

1. Method ‘canReceiveMessage(…)’: the default return value is ‘true’ even check fails, the default return value should be ‘false’.
2. Method ‘canReceiveMessage(…)’: after ‘if’ condition checking, should recipient be added to message(where message is modified?), something like “message.setAddRecipientAllowed(recipient, true)” before returning message.getAddReipientAllowed(recipient)?
3. Method ‘addRecipientToAllMessages(…)’: should it be written like below? There is a relation between recipient and succesfullMessages, it tells which recipient can receive those messages, I don’t see a relation between operator and messages or do I miss anything?

if (!successfullMessage.isEmpty())

recipientManager.addRecipient(recipient, successfulMessages);

Original test code:

1. The first assertEquals(message.size(), 2) check if Java ArrayList works, doesn’t touch client code.
2. Same user is passed into addRecipientToAllMessages(…), it only checks first condition, not the second condition when operator and recipient is different,
3. Currentuser is the same as recipient, canReceiveMessage(…) will return one true and one false, so it passes test.

Proposed test code:

1. Test case 1:

* User user3 = new User(“user3”);
* Message message = new Message(“test1”);
* Message.setAddRecipientAllowed(user3, false);
* Messages.add(message); // add new message to message list
* Result = rec. addRecipientToAllMessages(messages, user, user3); // it checks if user has right to modify recipient, recipient should be able to receive message, but original code won’t do that, so test fails

1. Test case 2:

* Message message3 = new Message(“test2”);
* Message.setAddRecipientAllowed(user3, false);
* Messages.add(message3); // add new message to message list
* Result = rec. addRecipientToAllMessages(messages, user, user3); // it checks if current user is the same as recipient, regardless of message Message.setAddRecipientAllowed(user3, false) is set false or true, user can be added to the recipient of message
* Original code fails

1. Test case 3:

* Create users and messages that there are successful message being added
* Check “recipientManager” to see if new user and its corresponding messages are added
* Test will fail.

**Specification review:**

1. **List of questions:**

* “As a user I can only edit my own profile”. Has it to be a writable user? Is it in conflict with the first statement “as a writable user….”? Probably it is beyond this question, how to create a writable user and non-writable user? When users create accounts, they can make it non-writable or somebody else like CSR can create ‘dummy’ accounts? I think I don’t really understand this part.
* “As a CSR I can edit any user” Can I assume CSR is a super user with all privileges? Can CSR edit others password? Should others get notification when password is changed? How should others know the password? It is not safe to send it directly using email. Can users reset password using ‘forget password’? If users choose ‘forget’ password to reset, it will nullify CSR’S effort to change user password.
* When password is reset by CSR, will it log out user? Can user still persist his profile data to back-end DB or need to re-login?
* Would deactivating user profile block editing? Should I assume not?

1. **Questions have higher priority.**

* Can CSR change password?
* If password is changed by CSR, how should user know the new password?
* Writable user and non-writable user?

1. **Reasons to choose those three questions.**

* Might need logic to create user accounts with different privileges.
* When two users edit profile at same time, we can assume whoever commits changes the last overrides previous one, but changing password is quite different. We might need new pages to handle this situation such as letting user know that password has been changed, you have to re-login to commit changes and so on.

1. **How will it affects development work**

* might need coding for a new page or alert and also add back-end logic to always check password for ‘saving profile’ event
* the first point will affect logic of who can edit user profile and introduce extra logic to create users with different privileges.