

Team 17 - WeGroup

Defect Log

Hanqi Du, Ao Liu, Xiaoyang Lin, Wanqi Jiang, Qinxin Shi, Zhixuan Yu

Product	WeGroup System Design Inspection		
Date	2/11/2016		
Author	WeGroup Team		
Moderator	Ao Liu		
Inspectors	Xiaoyang Lin, Hanqi Du, Zhixuan Yu, Qinxin Shi		
Recorders	Wanqi jiang		
Defect #	Description	Severity	How corrected
1	Server unable to handle requests from multiple clients at the same time	1	Server class extends to be multi-threaded
2	Time delay for too long between 2 users during chatting	2	Fetch the message from database instead of local files.
3	Server doesn't check if information of a user is already in the database when registering	2	Add checking blocks.
4	Server does not check if conversation already in the database	1	Add checking blocks
5	The "Forgot Password" button does not work	3	"Forget Password" was not hooked up with an function. Defect fixed by connecting the button to call performance of a segue

Product	WeGroup Mobile Client Design Inspection		
Date	2/7/2016		
Author	WeGroup Team		
Moderator	Qinxin Shi		
Inspectors	Hanqi Du, Ao Liu, Xiaoyang Lin, Zhixuan Yu		
Recorders	Wanqi Jiang		
Defect #	Description	Severity	How corrected
1	User uses "login with Twitter" option and after authorizing with twitter, the user cannot be brought back to WeGroup automatically	2	The app did not handle the callback url twitter sent back, so Safari tried to handle it by itself. We added the handle to the app and Safari will ask the user if he/she would like to go back. This fixed the problem.

2	The app does not successfully detect that a user has logged in before and automatically login for the user securely	3	The defect was caused by unsuccessful local storage of user id. We fixed the problem by using NSUserDefaults
3	The app automatically logs in for a user after he/she has logged out last time	2	The defect arose because the developer who wrote the code forgot to clear out the user id in local storage
4	After typing in the text field in Settings view, the keyboard does not disappear	2	We fixed the problem by adding a textFieldDelegate and manually configured when to dismiss keyboard

Product	WeGroup Code Inspection		
Date	2/8/2016		
Author	WeGroup Team		
Moderator	Hanqi Du		
Inspecters	Qinxin Shi, Xiaoyang Lin, Zhixuan Yu, Wanqi Jiang		
Recorders	Ao Liu		
Defect #	Description	Severity	How corrected
1	Group array index out of bound exception when new users join group	1	Use dynamic array, double array size when array is full.
2	App crashes when the username is of more than 14 characters	1	Add checking statement in the client
3	Password contains illegal characters	3	Add checking statement to prevent sending illegal input to backend
4	If “short bio” is more than 200 characters, the app does not perform as expected	2	Add checking statement for “short bio” in the client end
5	App crashes when user clicks “login” with email blank	2	Check whether email and password text fields are blank before sending information to server
6	Incorrect segue identifier from sign up view to main chat view	1	In most circumstances the user logs in from the loginView, therefore we forgot to test segue from signUpView. We fixed the problem by setting an identifier in storyboard and providing the corresponding identifier in code

Product	WeGroup Code Inspection		
Date	2/9/2016		

Author	WeGroup Team		
Moderator	Xiaoyang Lin		
Inspecters	Qinxin Shi, Hanqi Du, Ao Liu, Wanqi Jiang		
Recorders	Zhixuan Yu		
Defect #	Description	Severity	How corrected
1	Dialog text field is not being reset to empty after sending messages.	2	Set the text field to an empty string
2	Cannot sign in with correct email and password	1	Use equals() instead of “==”
3	User can send empty message	3	Disable send button when there is no input
4	Logout button has no response when clicking on it	2	Remove button and made the tableview cell to be the “logout button”. In the function that triggers when a cell is selected, check whether the cell indexPath equals the row the cell is in, if so perform logout.
5	The user can simply log in by clicking on “login” with empty email and password	2	The “login” button was called directly to perform segue for testers convenience
6	Unsuspected loading of user profile image	2	The function is supposed to set image directly to profileImageView, instead the it was set to imageView (the more general view) so it works but operates weird at times

Product	Login Unit Test		
Date	2/12/2016		
Author	WeGroup Team		
Moderator	Wanqi Jiang		
Inspecters	Hanqi Du, Qinxin Shi, Xiaoyang Lin, Zhixuan Yu		
Recorders	Ao Liu		
Defect #	Description	Severity	How corrected
1	Sign in succeeds with no email and password	2	Add checking statement to check the length of email and password
2	“login with Twitter” does nothing	1	The function gets request token from twitter but did not open the callback url successfully. Adding correct code to open url in Safari fixed the problem

3	User types in different inputs in “password” and “password confirmation” text boxes, but the app passes	1	Add checking whether the two text fields have the same input
4	Clicking login button nothing shows up to indicate user the signup status	2	In the back check if signup succeeded. If so segue into main chatting view, otherwise sends an alert view to indicate something is wrong
5	“Back to login” button does not work as it suppose	1	Fixed by giving the segue an identifier and perform the segue in the call
6	During signing in, if the input username or password starts or ends with space character, the sign in will be failed	2	Parse the username and password string before processing
7	When user open WeGroup after closing this app, he/she needs to re-login	3	add a judgement, if user does not log out, automatically login user each time he/she open this app.

Product	Sign up Unit Test		
Date	2/12/2016		
Author	Qinxin Shi		
Defect #	Description	Severity	How corrected
1	There are three text boxes for email, password and password confirmation. After typing in the first box, hitting return button does nothing	2	Manually configured to pass the user to the next text box until the user has filled in all three text boxes.
2	During signing up, if the format of the email address is incorrect (for example, there’s no ‘@’ character), hitting return button does nothing.	1	Show an error message until the email address is in a valid format.
3	During signing up, if the input email address or password starts or ends with space character, the expected email address cannot be signed up correctly	1	Parse the email and password string before processing
4	If the user sign up with a very short or very long password, the sign up will be successful without warning message	3	check the length of the password and make sure it’s not too long or too short
5	User can sign up with an already existing username	1	add notification ‘This username ’has already been used, please try another one’

6	No error message if the input of password is different from the input of password confirmation	2	Add error message to announce that the password is different from the password confirmation.
7.	Click forget password button and enter a email that has not been registered.	3	Show”This account does not exist”

Product	Chat Unit Test		
Date	2/12/2016		
Author	Ao Liu		
Defect #	Description	Severity	How corrected
1	Messages are not being sent to the other end on time	1	Correct the database storage for conversations when a new conversation is created by the user
2	When user creates a new topic, not everyone in the group can see the topic	1	In the backend, make sure to check and iterate through to put all members that are in the conversation in the array that receives features of being in a topic
3	When a topic is deleted, some group members still can see the topic	1	Fix backend to make sure clear all the related data of a topic when a topic is removed
4	User can send empty messages	2	Disable the “send” button when message text field is empty
5	App behaves abnormally when user tries to send very long messages	2	Add checking statement on the client end that a single message cannot be over 200 characters. If the case happens, the system should pop up an alert view and inform the user that the message is illegal
6	App crashes when the user tries to add new members to a group conversation when the group reaches upper limit	2	Add checking statement every time before the user adds friends to a group to see if a group has reached limit, if so pop up a warning to the user to inform that the group cannot have more than limited amount of people
7	A group remains when a user removes all the other users in the group	3	Check the number of people in the group every time the user adds or removes users. When the number becomes 2, the conversation becomes private, and when the number reaches 1, the system automatically deletes the conversation.