

## Defect Log (Sprint 2) - Team 4

Matthew Lenartowicz, Tyler Grikas, Aaron Dierking, Evan Tragesser, Guntas Grewal, Jacob Nordland

### Design Inspection Defect Log

<b>Product</b>	StudyBuddy Design Inspection
<b>Date</b>	3/05/2015
<b>Author</b>	StudyBuddy Team
<b>Moderator</b>	Evan Tragesser
<b>Inspectors</b>	Aaron Dierking, Matthew Lenartowicz, Guntas Grewal, Jacob Nordland
<b>Recorder</b>	Tyler Grikas

Defect #	Description	Severity	How Corrected
1	Sinch API does not take into account the fact that Parse users are split into individual study groups. It sends messages to all users in the database, rather than just the users in the same study group.	1	Query the Parse database within the Parse API to get just the users that are in the current user's study group.
2	If the Sinch API fails to receive the username of the user who just sent a message from the Parse server, it does not handle the error. Thus, the message is not received and no toast occurs.	2	Add a check in Sinch to check if the Parse query was successful. If it was not, display an error toast.
3	If the Sinch API fails to store a message in the Parse server, it is not handled. Thus, if a user re-opens the message log, the message will not be there.	2	Add a check in Sinch to check if the Parse query was successful. If it was not, retry storing the message 3x, and if it still fails, display an error toast.
4	If the Sinch API fails to load the message history from the Parse server, the error is not handled. Thus, nothing is displayed in the message log.	3	Add a check in Sinch to check if the Parse queries were successful. If they were not, display an error toast.
5	If getLocation fails to get your GPS location, it returns null. This causes other modules which depend on GPS (like request list) to crash.	1	Display a toast if getLocation fails, and edit the modules' code to check if it returned null.

### Code Inspection Defect Log

<b>Product</b>	StudyBuddy Code Inspection
<b>Date</b>	3/05/2015
<b>Author</b>	StudyBuddy Team
<b>Moderator</b>	Evan Tragesser

<b>Inspectors</b>	Aaron Dierking, Matthew Lenartowicz, Guntas Grewal, Jacob Nordland
<b>Recorder</b>	Tyler Grikas

<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How Corrected</b>
1	Sinch toasts should be resource strings, rather than hard-coded strings.	3	Changed the hard-coded strings into resource strings.
2	If a message fails to send, it does not display an error to the user. It only puts an error in the debug log.	1	Added a toast which informs the user if the message fails to send.
3	When a request is selected, we save the request title in the bundle, but this is redundant because all we need is the request id, which we also save.	3	Removed code that saves the request title.
4	When creating a new request, the "create" button is not locked. Thus, the user can rapidly press the button which creates duplicate requests, as well as displays errors.	1	Lock the "create" button after tapping it once, and unlock it if creating the request fails.
5	My Group page does not check if the user's group has been deleted. The group info page is displayed as if it still exists.	2	Query the Parse database with the group's request ID and check if it is null. If it is, display a toast and clear the group info screen.
6	The request info page allows the user to join a group that has no time left (if the request list is loaded before the request timer runs out). This could cause a crash in the app and/or server code.	1	When a user tries to join a group, check that the time remaining is greater than zero.
7	In the Parse CloudCode function RequestTimer, we sort the requests by time remaining, but this is unnecessary because we loop through each request individually, decrement the timer, and check if its zero.	3	Remove the code that sorts the requests by time remaining.
8	In RequestListActivity, the code that compares two distances can be condensed into one line of code, rather than 6 lines. This could be slightly more efficient.	3	Change 6 lines of code into the slightly more efficient one line of code.
9	In ClassAddActivity, the code that adds a class locally should be placed in its own method, similar to how removing a class has its own method.	3	Put the code for adding a class in its own method.
10	Whenever the Messages tab is accessed, all messages are loaded from the server. This is very slow and thus inconvenient for the user.	1	Cache all previously received messages, so when the Messages tab is accessed, only the new messages need to be loaded.
11	After joining or leaving a group, the member and helper count is not updated immediately.	2	Added code that updates the UI once you leave or join a group.
12	Help Requests don't refresh after the user makes a new help request and returns to the RequestListActivity.	2	Added a return intent to the activity to refresh the request list.

13	Request info page displays helper and user count in a way that makes it seem like "members" is not "total members".	3	Changed the UI to be "X members (Y helpers)", where X and Y are numbers. It is a bit clearer that helpers are a subset of members.
14	Help Requests do not refresh automatically after a certain amount of time	2	Added a handle that runs approx. every 30 seconds, which will auto-update the help request list.
15	Push notifications for new help requests are sent to the user that created them.	2	Added condition to Parse Query to exclude the user that created the help request.

## Unit Testing Defect Log

<b>Product</b>	StudyBuddy Unit Test
<b>Date</b>	3/2/2015
<b>Author</b>	StudyBuddy Team

Defect #	Description	Severity	How Corrected
1	Sinch messages were getting sent to the current user and causing an error to be printed to the debug console.	3	Adjusted user query to ignore the current user.
2	Messages were showing the user's object ID instead of the user's display name. This is a security issue because object IDs are internal values.	1	Query Parse to get the username each time a message is received.
3	When you click on a help request push notification while the app is open, you get the message "Unfortunately Studdy Buddy has stopped," and the app crashes.	1	Added an activity to handle push notifications, and a specific method that handles them when the app is open.
4	When you click on a help request push notification without the app open, you get the message "Unfortunately Studdy Buddy has stopped". The app does not start.	1	Added an activity to handle push notifications, and a specific method that handles them when the app is open.
5	When Request Info fragment is first loaded, it displays garbage text until it is replaced by actual text loaded from the server.	2	Added a loading indicator that prevents the user from seeing these garbage values.
6	Cached member/helper count of the current group (immediately after joining the group) is missing the user that just joined.	3	Incremented the member (and helper, if the user joined as one) counts before caching them.
7	When you try to access My Group page, the app crashes. Console indicates a null pointer exception.	1	Changed the parse query to retrieve the numbers which indicate the number of helpers and members, rather than retrieving strings.
8	Cached helper count is incorrectly being displayed as 0, and member count is always 1.	2	Changed the way we store helper/member count. User has 2 fields in the database which stores these numbers.

9	After leaving a group at the My Group page, if you tap My Group in the navigation drawer, it should go to the "No Group" page. Instead, it goes to the group's info page.	2	Added a method that refreshes the page; is called when the group page is selected.
10	If you select My Group from the navigation drawer with no network, the app crashes.	2	Added a Parse callback method which ensures that if there is network failure, the exception is caught and the app doesn't crash.
11	If you leave the group you are currently in from the My Group page, and then join it again from the same page, the app crashes.	2	Group-joining code used a variable (the group's request object) that was only set if the info page was accessed through the request list. Now that variable is set to the user's current group's request object when viewed through My Group
12	If you are on the Messaging tab typing a message, and you choose to make the keyboard disappear by tapping the screen, the keyboard does not go away.	3	Explicitly enabled this Android OS feature in the Messaging activity.
13	Request list displaying path of object in java code, rather than the actual title of the request.	1	The android GUI component uses the class's default "toString" method, so we overrode it for the ClassRequest class and have it return the title of the request.
14	If you back out of the Request Info page while it is still loading, you get the error "Unfortunately Study Buddy has stopped," but the app continues to run.	3	Added code in the onBackPressed method which cancels the Parse request, if it is still loading.
15	If you create a new request but the app cannot get GPS signal (even if you have wifi), it will toast saying the request was successful, but then toast that it cannot get your location. The request list does not then auto-update, even though the request was actually created.	2	Moved the code which saves the request in the Parse database after the location is successfully retrieved.
16	If you change tabs from the messages screen and then revisit the messages screen, the list of old messages is not in the right order.	1	Synchronized the two queries which run parallel to wait for one another to finish before ordering the messages.
17	If you join a group, then leave it, then click on the my group tab on the drawer, the loading popup will be there indefinitely.	1	Moved the code which dismisses the loading indicator, to a spot which properly closes it.
18	When you send a message, it often sends multiple duplicate messages instead of one (a range from 1-4).	1	Lock the 'Send' button after it is first pressed, and unlock it once the message successfully sends or fails to send.