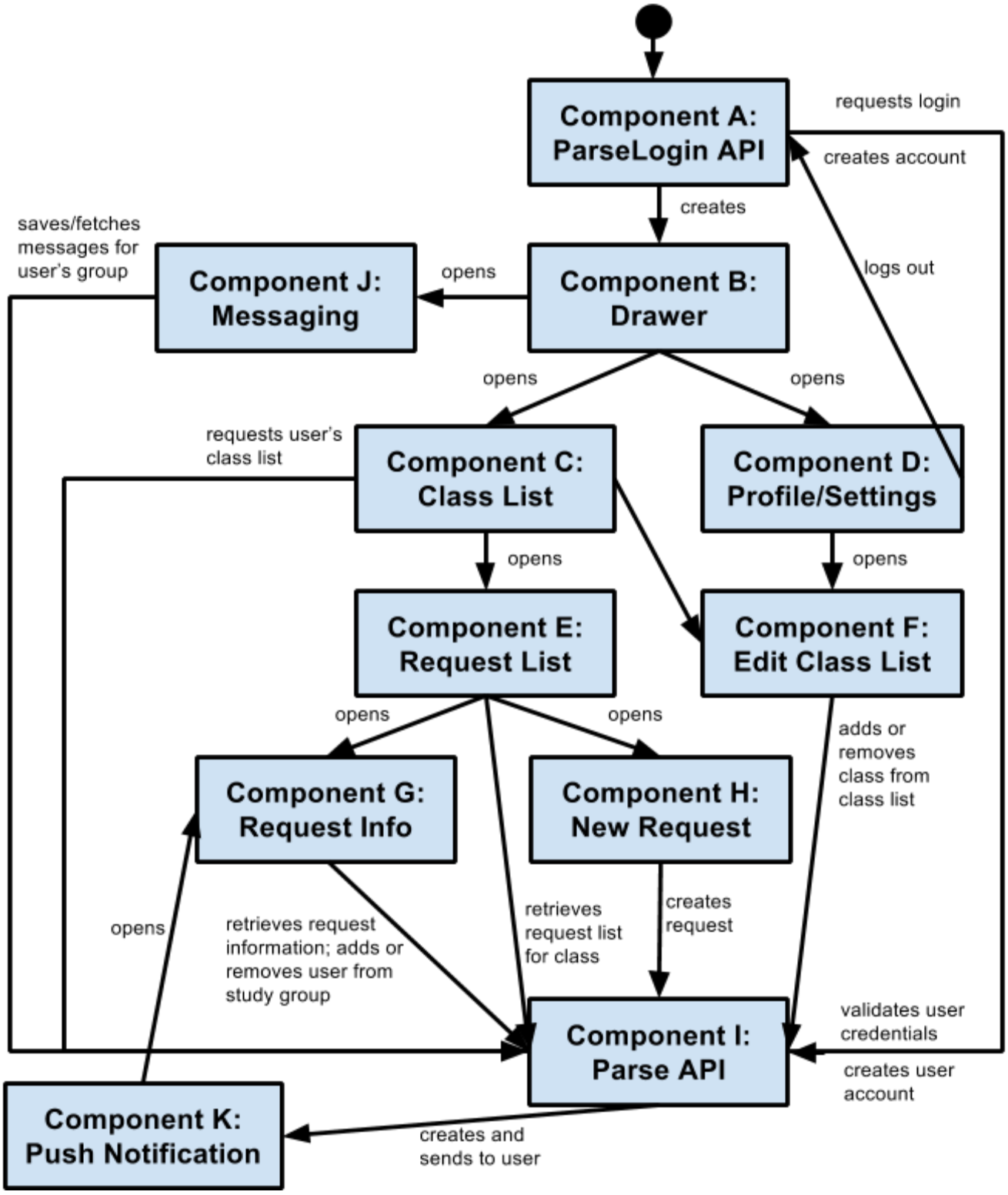
Incremental and Regression Testing - Sprint 2 - Team 4

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

Component Definitions

Component Id	Component Name	Definition
A	ParseLogin API	UI that interacts with the Parse API to allow users to log in, and create new accounts.
B	Drawer	A slide-in menu where the user can choose which section of the app they want to navigate to. (e.g. Classes, My Group, etc.)
C	Class List	The list of classes the user has selected to be in. Tapping on a class brings you to the Request List for the specific class.
D	Profile/Settings	Page where the user can log out of their account, turn off push notifications, and add/remove classes from their class list.
E	Request List	The list of active requests for a specific class. Tapping on a request will bring up the Request Info page for the specific request.
F	Edit Class List	Page where the user can add or remove classes from their class list.
G	Request Info	Page that displays information about a selected request, and allows the user to accept the request.
H	New Request	Page to create a new request, in a class specified by the user. The user must fill out information about the request, which can be viewed on the Request Info page after it is created.
I	Parse API	API that is used for interaction between the application and the database/server. It is used to store user login information, classes, user class lists, and requests.
J	Messaging	Page which allows users to send text-based messages to one another.
K	Push Notification	Generated by Cloud Code in the Parse API, push notifications are sent to the user's device when a nearby student creates a request for a class the user is also in.

Component Diagram



Input, Output and the Dependent Components

Component Id	Component Name	Possible Inputs	Outputs	Dependent Components
A	ParseLogin API	**(a)***Logging in:* Username and password, typed into text boxes using the Android keyboard. **(b)***Creating an account:* Username, Password, Confirmed Password, Email Address (typed into text boxes using Android keyboard).	**(a)***Logging in:* Checks with the Parse database if the username and password combo is valid or invalid. If invalid (or network error occurs), display an error. If valid, the user is logged in and the app navigates to the Class List. **(b)***Creating an account:* Automatically logs in if account creation is successful, and the app navigates to the Class List. Displays error if not successful (if there are invalid fields, or a network error).	B, I
B	Drawer	Touch input that selects the page the user wants to navigate to.	Navigates the user to the selected page.	A, C, D
C	Class List	Touch input that selects the class the user wants to view/add requests for.	Navigates the user to the Request List page for the selected class.	B, E, I
D	Profile/Settings	**(a)**Touch input that would register as a button press to either: Log out, **(b)**Turn off notifications, or **(c)**Edit the user's class list.	**(a)**Logs the user out and navigates them to the login page, **(b)**Turns off notifications, **(c)**Navigates the user to the edit class list page.	A, B, F, I
E	Request List	**(a)**Touch input that selects which request the user wants to view the Request Info page for, or **(b)**Allows the user to	**(a)**Navigates the user to the Request Info page for the selected request, **(b)**Navigates the user to the New Request page.	C, G, H, I

		create a new request.		
F	Edit Class List	**(a)**Text field where the user can type in a class name to narrow down a list of all classes available to add to the user's class list. **(b)**The user selects, by touch, a class to add to their class list.	**(a)**The list is automatically updated to contain just the classes that match the text currently in the text field. If no classes match, the list is empty. **(b)**Class is added to the user's class list. This is updated in the UI, local files, and in the Parse database. A toast message appears indicating success (or failure, if there is a network error).	C, D, I
G	Request Info	**(a)**Touch input that would register as a button press to accept a help request (and thus join the request's study group), or **(b)**leave the study group, if the user is already in the group he/she is viewing. *An alert box appears after touching the accept/leave button, that has a "Yes" button and "Cancel" button (which takes touch input on the buttons).*	**(a)**If the user is already in a study group, the alert box appears. If "Yes" is selected, the box disappears and the user is removed from the previous group (both locally and in the Parse database). If "Cancel" is selected, the box disappears and nothing happens. If "Yes" is selected or the user is not already in a group, the user is added to the request's study group both locally and in the Parse database. The button is updated to read "Leave Group". **(b)**The alert box appears. The behavior is the same as above, except the user does not join a new group. The button is updated to read "Join Group".	E, I
H	New Request	**(a)**Several text boxes for entering in information: *Title, Location,* and *Description* fields to let people know what is being worked on, and where. Text is entered using the Android keyboard. There are also two drop-downs to select the amount of	**(a)**As the user types in the text boxes, characters will automatically stop appearing when the character limit for the specific text box is reached. **(b)**If the button is pressed and any of the text fields are empty and/or the time is set to zero, there will be a toast	E,I

		time (*Duration*) the user is planning to work on the assignment. **(b)**Finally, there is a *Create* button to create the request when the user has finished entering the above fields (touch input).	pop-up that notifies the user what is missing, and the app stays on the page. If the button is pressed and all fields are valid, the app attempts to create the request locally and in the Parse database. If there is an error creating the request, the user will be notified of a network error. If creating the request succeeds, a toast message indicating success appears and the app goes back to the Request List page.	
I	Parse API	**(a)***Login*: Username, Password, Device ID **(b)***Log out:* Username, Device ID **(c)***Account Creation*: Username, Password, Email address. **(d)***Class List:* Username. **(e)***Request List:* Class name. **(f)***Add Class:* Class name, Username. **(g)***Remove Class:* Class name, Username. **(h)***Join Group:* Request ID, Username **(i)***Leave Group:* Request ID, Username **(j)***Create Request:* Title, Location, Description, Duration, Username **(k)***Retrieve Messages*: Request ID **(l)***Save Message:*	**(a)**Notifies app if username and password combo is valid. If valid, associates user with the installation object (the user's device) in the database. **(b)**Removes the association between the user and the installation object in the database. **(c)**Creates the account on the database, and specifies if account creation was successful. **(d)**Fetches the database object for each class the user is in. Notifies success or failure. **(e)**Fetches the database object for each active request for the specified class. Notifies success or failure. **(f)**Adds the class to the user's class list in the database. Notifies success or failure. **(g)**Removes the class from the user's class list in the database. Notifies success or failure.	A, C, E, F, G, H

		Request ID, Message Text	**(h)**Adds the user to the study group associated with the request, in the database. Notifies success or failure. **(i)**Removes the user from the study group associated with the request, in the database. Notifies success or failure. **(j)**Creates a new request in the database. Notifies success or failure. **(k)**Retrieves all messages that have been sent in a particular study group. Notifies success or failure. **(l)**Saves message in the database, tied to the study group. Notifies success or failure.	
J	Messaging	Text field where the user can type, using the Android keyboard, the message they wish to send to their group members. Tapping the “Send” button indicates the user wants the message to be sent.	Message text is sent to all group members, and appears in everyone’s chat log. The message appears differently if it was sent by the user who is viewing it. Uses Component I (Parse API) to save the message on the server.	B, I
K	Push Notifications	The user can tap the notification which appears in the Android OS’s notification area.	Component G (Request Info) is opened for the specific request that generated the push notification that was tapped.	G, I

Form of Incremental Testing

The form of incremental testing we followed is **bottom-up**. We chose this because nearly every module depends on the Parse API, so testing it alone first and then testing each module with its functionality makes the most sense for our application. We tested the components in the following order: *I, H, G, F, E, J, D, C, B,* and then *A*.

Incremental and Regression Testing

Module	Component A - ParseLogin API

Incremental Testing

Defect #	Description	Severity	How To Correct
1	The logo displayed on the login page is for Parse itself, not StudyBuddy.	3	create a new image and replace the current one with it.
2	The “Log in” button is not user-friendly; it is covered when the user types in their email and password. This is confusing and inconvenient for the user.	3	Place content in scroll view so you don’t have to close the keyboard

Regression Testing

Defect #	Description	Severity	How To Correct
1	No additional defects were discovered through regression testing of this component, in its current state.	N/A	N/A

Module	Component B - Drawer

Incremental Testing

Defect #	Description	Severity	How To Correct
1	Opening the “My Group” (Component G) tab from the drawer doesn’t refresh the page.	2	Moved the request info (Component G) refresh code into a method and called it when selecting “My Group”

2	Opening “My Group” (Component G) tab with no internet connection causes the app to immediately crash.	2	Added a check for internet connection so it can quickly load the cached copy without crashing (or spending ~20 seconds querying the database)

Regression Testing

Defect #	Description	Severity	How To Correct
1	No additional defects were discovered through regression testing of this component, in its current state.	N/A	N/A

Module	Component C - Class List

Incremental Testing

Defect #	Description	Severity	How To Correct
1	There is no loading indicator for the class list; when it is actually loading, it appears that the user just doesn’t have any classes registered.	3	Add a loading indicator while the app is looking up the user’s class list.

Regression Testing

Defect #	Description	Severity	How To Correct
1	No additional defects were discovered through regression testing of this component, in its current state.	N/A	N/A

Module	Component D - Profile/Settings

Incremental Testing

Defect #	Description	Severity	How To Correct
1	No defects currently. This module is currently only used as a path to Component A (ParseLogin API) and Component F (Edit Class List). It doesn't have any of its own functionality.	N/A	Add this module's functionality (edit user profile, silence notifications).

Regression Testing

Defect #	Description	Severity	How To Correct
1	No defects currently. This module is currently only used as a path to Component A (ParseLogin API) and Component F (Edit Class List). It doesn't have any of its own functionality.	N/A	Add this module's functionality (edit user profile, silence notifications).

Module	Component E - Request List

Incremental Testing

Defect #	Description	Severity	How To Correct
1	Returning to the request list after making a new request (Component H) or viewing a request (Component G) made the location feature stop working and return null.	1	Start the location service in onResume as well as onCreate so it can start the gps again when returning.
2	There is no indication that the request list is loading; the page is simply blank while it loads.	3	Add a loading indicator while the application looks up requests.

Regression Testing

Defect #	Description	Severity	How To Correct
1	Fixing defect #1 of this component caused a new defect: If you create a new request (in Component H), the app will toast saying the request was successful, but then toast that it cannot get your location. The request list is not updated, even though the request was created.	1	Removed call to stop the gps in the onDestroy method in the NewRequest activity(Component H). This caused a race condition that would stop the gps after this activity started it.

Module	Component F - Edit Class List

Incremental Testing

Defect #	Description	Severity	How To Correct
1	When you add or remove a class, there is no loading indicator, which makes the process very confusing. It is especially difficult to tell that you've successively tried to add a class.	2	Add a loading indicator to inform the user what is happening during the period while the application is adding / removing their class.
2	When adding and removing classes, the class list is pulled from the local cached version on the device instead of from the database.	2	Make the class list be pulled from parse instead of from the device.

Regression Testing

Defect #	Description	Severity	How To Correct
1	No additional defects were discovered through regression testing of this component, in its current state.	N/A	N/A

Module	Component G - Request Info

Incremental Testing

Defect #	Description	Severity	How To Correct
1	It does not check if the user's group has been deleted. The request info page is displayed as if it still exists.	2	Added a query if the user has a network connection that checks if the request exists. If it no longer exists the app now automatically leaves the request and then displays a message to the user.
2	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	Added a query for the request before attempting to join. This checks if the request still exists.
3	If the page attempts to load with no internet connection, it loads for ~10 sec and the activity crashes. The app remains running.	3	When loading a new request, moved the finish() call into the UIThread section, and returned null in the list instead. For the My Group page it now checks for a connection before trying to update the member count so it can skip the parse query.
4	Quickly pressing any of the join or leave buttons updates the member count multiple times	2	Added a boolean value to check whether the app is still performing tasks from the previous button press before allowing a new press.
5	Dummy info can be seen before the actual page loads	2	Added a loading icon to show that data is being

			downloaded and hides the view until it is ready.

Regression Testing

Defect #	Description	Severity	How To Correct
1	After fixing issue 5, loading the My Group without being in a group or after an error would make the loading wheel persist indefinitely	3	Added a call to dismiss the loading icon when reaching these two cases

Module	Component H - New Request

Incremental Testing

Defect #	Description	Severity	How To Correct
1	Pressing the “Create” button quickly will submit the request several times	2	Added a boolean to check whether the app was still in the process of submitting a request before allowing another attempt.

Regression Testing

Defect #	Description	Severity	How To Correct
1	No additional defects were discovered through regression testing of this component, in its current state.	N/A	N/A

Module	Component I - Parse API

Incremental Testing

Defect #	Description	Severity	How To Correct
1	Parse is repeatedly generating push notifications for the same help request. This is incredibly annoying and inconvenient for the user.	1	Set a flag for when a user has a unique notification, and only allow one unique notification per group to be sent to the user.

Regression Testing

Defect #	Description	Severity	How To Correct
1	No additional defects were discovered through regression testing of this component, in its current state.	N/A	N/A

Module	Component J - Messaging

Incremental Testing

Defect #	Description	Severity	How To Correct
1	If a message fails to send, it does not display an error to the user. It only puts an error in the debug log.	2	Made a toast display when a message fails to send.
2	Whenever the Messages tab is accessed, all messages are loaded from the server. This is very slow and thus inconvenient for the user.	2	Implemented a system which caches messages to a local database so that the tab loads faster.
3	If you are typing a message, and you choose to make the keyboard disappear by tapping the screen, the keyboard does not go away.	3	Used onTouchEvent() to hide the keyboard when user selects an area on

			the screen other than the keyboard.
4	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	This was caused by a race condition. By fixing the race condition and enforcing a sorted list, this issue was resolved.
5	When you send a message, it often sends multiple duplicate messages instead of one (a range from 1-4).	1	Adjusted the message sending code so that it sends one message to all recipients at once instead of sending a different message to each recipient.
6	Messages take a long time to send and hang the UI.	2	Implemented a feature where messages are shown in the UI immediately after pressing the Send button, and sending them is deferred.
7	Sending a message to no recipients crashes the application.	1	Modified the message sending code so it doesn't attempt to send anything if there are no recipients, and instead just stores the message in the database.
8	Sending a message to more than 10 recipients at a time causes the application to crash.	1	This is an issue with the 3rd-party library we use to send messages. The only way we know of to work around this right now is to send messages to groups of 10 people at a time.

Regression Testing

Defect #	Description	Severity	How To Correct
1	Fixing defect #6 created an issue where sending a message and then immediately switching to another tab	1	Keep the messaging code alive until all

	caused the message to not be sent completely.		messages have sent completely.
2	Fixing defect #6 made it difficult to tell when a message has finished sending completely.	3	Added a “Sending” indicator to messages that are in the process of being sent.
3	Fixing defect #2 caused the UI to lag slightly when loading the message list.	2	Made message history load in a background thread.

Module	**Component K - Push Notifications**

Incremental Testing

Defect #	Description	Severity	How To Correct
1	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	Created our own “ParsePushBroadcastReceiver” class and overrided the “onPushOpen” method.
2	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	Created our own “ParsePushBroadcastReceiver” class and overrided the “onPushOpen” method.

Regression Testing

Defect #	Description	Severity	How To Correct
1	After fixing defect #1 and #2, clicking on a push notification directs the user to the app’s splash screen, if the app isn’t open. If the app is open and was opened originally from a push notification, it simply brings the application to the foreground, which is good. However, if the app was originally opened via	2	Made onPushOpen open LoginActivity (Component A) rather than the Splash screen. Due to auto-login, this works perfectly.

	clicking its icon, clicking a push notification opens the splash screen activity, which is bad.		