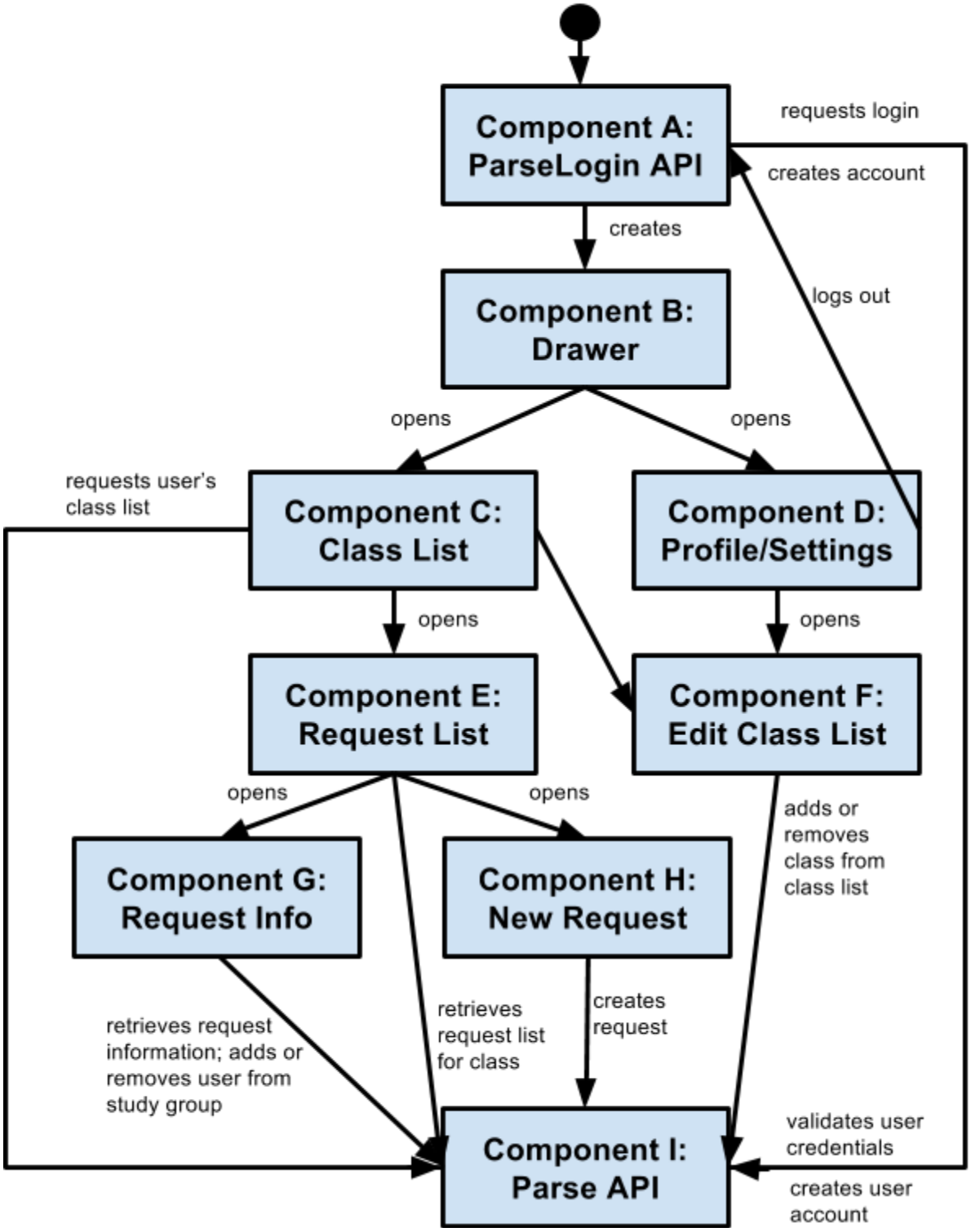
Incremental and Regression Testing - Team 4

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.

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 create a new request.	**(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
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	**(a)**The list is automatically updated to contain just the classes that match the text currently in the text field. If no	C, D, I

		the user's class list. **(b)**The user selects, by touch, a class to add to their class list.	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).	
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 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).	**(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 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	E,I

			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	**(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. **(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.	A, C, E, F, G, H

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, D, C, B,* and then *A*.

Incremental and Regression Testing

Module	**Component A - ParseLogin API**

Incremental Testing

Defect #	Description	Severity	How To Correct
1	When typing in your email and password, the on-screen keyboard covers up the Login and Sign Up buttons.	3	Adjust the UI so that the buttons move to be above the keyboard when it is opened.
2	Pressing the Android OS ‘back’ button takes the user to a blank page.	3	Have the back button close the application.
3	Signing up for an account without an internet connection causes a “Loading” message to display for a long time (~20 sec) before showing the user an error.	3	Detect if the user has internet connectivity instead of waiting for the request to time out.

Regression Testing

Defect #	Description	Severity	How To Correct
1	When fixing defect #2 of this component, we kill the ParseLogin API thread. This crashes the application, instead of simply closing it without an error message.	3	Properly utilize the Android onDestroy() method so the application doesn’t crash.

Module	Component B - Drawer

Incremental Testing

Defect #	Description	Severity	How To Correct
1	If you try to access Component D (Profile/Settings) from the drawer directly from this component, sometimes under unknown conditions, the class list is not cleared from the view and is superimposed over the profile/settings page. All buttons are still usable, so this is a minor graphical glitch.	3	Check in class list and drawer activity to see if we are clearing the view / context properly when switching to the profile/settings page.

Regression Testing

Defect #	Description	Severity	How To Correct
1	Attempting to fix the issue where two menus appear at once (defect #1 of this component) by explicitly clearing all views created an issue where picking the same view that the user is already in would clear the view and leave a blank screen.	1	Delay removing all views when switching views via the navigation drawer until it has passed the check for whether the user has picked the same view he/she is already in.

Module	Component C - Class List

Incremental Testing

Defect #	Description	Severity	How To Correct
1	When the app is opened for the first time the user will not have any classes and there was no indication on how to add classes.	2	If the user has no classes add a button to bring him/her straight to Component F (Edit

			Class List) and a message on where to edit classes in the future.
2	While the class list loads, it displays a “No Classes” message instead of a loading indicator.	3	Display a loading indicator instead while the class list loads
3	If a class is selected when the user does not have an internet connection, the application crashes.	2	Temporarily remove code that will be used in a later feature (GPS location).

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	There is no loading indicator while the request list loads. This becomes more of an issue when the user loses internet connection, since nothing will happen for approx. 15 sec (although the app doesn't freeze), and then our network error toast message finally appears.	3	Display a loading indicator while the request list loads.
2	If you back out of the request list while the list is still loading, and then select the same or different class, the application crashes.	1	This was fixed when we fixed Component C (Class List)'s defect #3.
3	If you navigate to Component H (New Request) while the request list is still loading, attempting to submit a new request will cause the app to crash.	1	This was fixed when we fixed Component C (Class List)'s defect #3.

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 F - Edit Class List

Incremental Testing

Defect #	Description	Severity	How To Correct
1	Users could enter in anything into the class textbox, even if it wasn't a real class. If they tried to add this invalid class, the application would crash.	1	Include a list of classes and have the user select from that list, rather than type it in.
2	Attempting to add a class without an internet connection displays an "Error: Class Not Found" message instead of an internet connection error message.	3	Check if errors occur as a result of a loss of internet connectivity, and display an appropriate message.
3	When you add a class, it adds it locally immediately, while the class is added to the server in the background. If the server interaction fails, it is automatically removed locally. However, if you manually remove it locally before the server interaction fails, the app crashes when it tries to automatically remove it.	2	Wait for the server interaction to succeed before adding the class locally. Alert the user that it is loading by a message or a loading indicator.

Regression Testing

Defect #	Description	Severity	How To Correct
1	By loading the list of classes used to prevent entering values that aren't real classes (fixing defect #1 of this component), the page for editing classes began to load slowly.	3	Move loading the class list to a background task.

Module	Component G - Request Info

Incremental Testing

Defect #	Description	Severity	How To Correct
1	Forms that had put a lot of text into the location or description text boxes could push information off of the bottom of the screen, making them impossible to read.	1	Place the page into a scrollview so that all of the form's items can be reached.

Regression Testing

Defect #	Description	Severity	How To Correct
1	Using a scrollview allowed the toolbar at the top of the screen to scroll as well, which could push it off of the top of the screen, making it inaccessible to the user.	2	Move the toolbar to a view outside of the scrollview and place it at the top.

Module	Component H - New Request

Incremental Testing

Defect #	Description	Severity	How To Correct
1	Entering a lot of text into the textbox could push the lower items in the form off of the page, making them inaccessible to the user.	1	Place the page into a scrollview so that all of the form items can be reached at all times.
2	Creating a new request without an internet connection causes the app to freeze for a long time (~15 sec), and then finally display an error message.	2	Detect if the user has internet connectivity before attempting to create a request. Add a loading indicator in case connection is just slow.
3	The title textbox does not appear to scroll correctly on some devices. When the textbox overflows, it scrolls so that none of the previously entered text is visible.	3	Make the scrolling behavior the same as the Location and Description fields, which expand downward when the text overflows.

4	If you rapidly press the “Create” button, you are able to create multiple of the same request on the server.	2	Disable the “Create” button after it is pressed. (re-enabling it if creation fails)

Regression Testing

Defect #	Description	Severity	How To Correct
1	Using a scrollview allowed the toolbar at the top of the screen to scroll as well, which could push it off of the top of the screen, making it inaccessible to the user.	2	Move the toolbar to a view outside of the scrollview and place it at the top.

Module	**Component I - Parse API**

Incremental Testing

Defect #	Description	Severity	How To Correct
1	Accepting a request (joining a study group) returns success, but nothing actually changes in the database.	1	Change the “currentRequest” column in the User object (in the database) into a pointer, instead of a relation.

Regression Testing

Defect #	Description	Severity	How To Correct
1	After fixing defect #1 of this component, the application would crash when attempting to fetch the user’s current request (study group).	1	Access the “currentRequest” column with a different method, since pointers and relations require different semantics.