Testing:

Matthew: The tests that I ran were done to ensure that the basic functions of creating an account and creating and joining groups were functioning correctly. The equivalence class are proper inputs for account/group creation and group joining. I used a valid email and password to create my account and a valid group name and password to create a group. I then used the set group name and password to join the group. The expected results were that I received confirmation at the creation of the account and the group as well as when I joined the group. These expected results were also the ones I received.

Rishab: I tested proper input syntax for chores. The equivalence class is a correct input, which meant all fields were filled out. I entered a name for the chore as well as a name for the person assigned to said chore. The result was I received confirmation that the post was valid and it was immediately visible on the page, which is the expected result.

Ashvini: The test cases I created were made to test valid inputs for the expenses function. The equivalence class are valid inputs for the expenses, which is a name of a group member, a name for the chore, and a number value for the expense. The expected result of this is confirmation that the expense entry is valid and it post to the page immediately. The equivalence class yielded the expected results.

Sukhveer: I focused on the groups module. My test cases were focused around edge cases and 'bad' input. The equivalence classes are no input and bad input. For no input, I tried to create a group with no name, with no password, and with both no name and no password. For bad input, I created a group name with multiple words and special characters. The expected result for no input and bad input was to let the user know that the input was not correct, and ask them to try again. For both equivalence classes, the expected result occurred.

Rishika: My test cases were focused on edge cases and 'bad input' of the chores module. Its equivalence classes are no assigned person for the chore, or a person who doesn't exist is assigned the chore. For no assigned person, I tried to assign a chore to nobody. For a person who doesn't exist, I created a group and assigned a chore to a person who wasn't in the group. The expected result for no assigned person and the nonexistent person is to let the user know that the inputted value for the person is incorrect and to try again. For both equivalence classes, the expected result occurred.

Vinay: I focused on edge cases and bad input for the expenses module. The equivalence classes for this are no input, invalid number for expense, and nonexistent person assigned. For no input, I tried to create an expense with no value, no assigned person, no description, and no value, assigned person, and description. For an invalid number for an expense, I created an expense with the amount as '78a9'. For the nonexistent person assigned the expense, I created an expense for someone who wasn't in my group. The expected result for no value, invalid number for expense, and nonexistent person assigned to the expense was to let the user know that the input was not correct, and ask them to try again. For both equivalence classes, the expected result occurred.