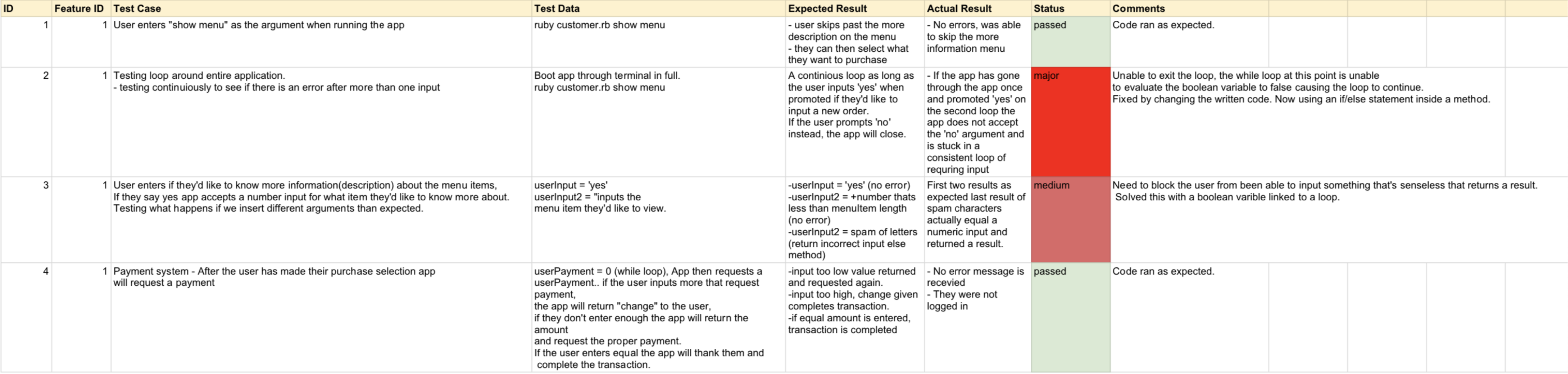
ID	Feature		
1	User has the ability to pass an information flag on the command line to skip the "more information" about burgers		
2	User has the ability to loop through majority of the app via a prompt at the end of each transaction. If they enter 'yes' app should loop, if they enter 'no' app should send a goodbye message and terminate		
3	User will be prompted during the application if they'd like further descriptions on the menu options, if they input 'yes' when prompted they will have a numbered list of options to chose from. Testing functionality and breakability of this section of the app.		
4	User will be prompted to make a payment for their menu selection, testing stability of this section of the app		



ID	Feature ID	Test Case	Test Data	Expected Result	Actual Result	Status	Comments
1	1	User enters "show menu" as the argument when running the app	ruby customer.rb show menu	- user skips past the more description on the menu - they can then select what they want to purchase	- No errors, was able to skip the more information menu	passed	Code ran as expected.
2	1	Testing loop around entire application. - testing continuiously to see if there is an error after more than one input	Boot app through terminal in full. ruby customer.rb show menu	A continious loop as long as the user inputs 'yes' when promoted if they'd like to input a new order. If the user prompts 'no' instead, the app will close.	- If the app has gone through the app once and promoted 'yes' on the second loop the app does not accept the 'no' argument and is stuck in a consistent loop of requring input	major	Unable to exit the loop, the while loop at this point is unable to evaluate the boolean variable to false causing the loop to continue. Fixed by changing the written code. Now using an if/else statement inside a method.
3	1	User enters if they'd like to know more information(description) about the menu items, If they say yes app accepts a number input for what item they'd like to know more about. Testing what happens if we insert different arguments than expected.	userInput = 'yes' userInput2 = "inputs the menu item they'd like to view.	-userInput = 'yes' (no error) -userInput2 = +number thats less than menuItem length (no error) -userInput2 = spam of letters (return incorrect input else method)	First two results as expected last result of spam characters actually equal a numeric input and returned a result.	medium	Need to block the user from been able to input something that's senseless that returns a result. Solved this with a boolean varible linked to a loop.
4	1	Payment system - After the user has made their purchase selection app will request a payment	userPayment = 0 (while loop), App then requests a userPayment.. if the user inputs more that request payment, the app will return "change" to the user, if they don't enter enough the app will return the amount and request the proper payment. If the user enters equal the app will thank them and complete the transaction.	-input too low value returned and requested again. -input too high, change given completes transaction. -if equal amount is entered, transaction is completed	- No error message is recevied - They were not logged in	passed	Code ran as expected.