

Cisc 327 Assignment 5

Test Cases:

Specification	Test Case ID	Purpose
The name of the ticket has to be alphanumeric-only	R4.1.1	Check if the selling actions succeed when the ticket names is alphanumeric-only
The name of the ticket has to be alphanumeric-only	R4.1.2	Check if the selling actions fail when the ticket names contains special characters
The name of the ticket can only have a space if it is not the first or last character	R4.1.3	Check if the selling actions succeed when the ticket name uses a space somewhere other than the beginning or end
The name of the ticket can only have a space if it is not the first or last character	R4.1.4	Check if the selling actions fail when the ticket name uses a space at the beginning or end
The name of the ticket is no longer than 60 characters	R4.2.1	Check if the selling actions succeed when the ticket name is 60 characters or less
The name of the ticket is no longer than 60 characters	R4.2.2	Check if the selling actions fail when the ticket name is greater than 60 characters
The quantity of the tickets has to be more than 0, and less than or equal to 100.	R4.3.1	Check if the selling actions succeed when the ticket quantity is greater than 0 and less than 101

The quantity of the tickets has to be more than 0, and less than or equal to 100.	R4.3.2	Check if the selling actions fail when the ticket quantity is 0 or less
The quantity of the tickets has to be more than 0, and less than or equal to 100.	R4.3.3	Check if the selling actions fail when the ticket quantity is greater than 100
Price has to be of range [10, 100]	R4.4.1	Check if the selling actions succeed when the ticket price is between 10 and 100
Price has to be of range [10, 100]	R4.4.2	Check if the selling actions fail when the ticket price is less than 10
Price has to be of range [10, 100]	R4.4.3	Check if the selling actions fail when the ticket price is greater than 100
Date must be given in the format YYYYMMDD (e.g. 20200901)	R4.5.1	Check if the selling actions succeed when the ticket date is in the correct format
Date must be given in the format YYYYMMDD (e.g. 20200901)	R4.5.2	Check if the selling actions fail when the ticket date is not in the correct format
For any errors, redirect back to / and show an error message	R4.6	Check that the current page is / and that an error message has been displayed
The added new ticket information will be posted on the user profile page	R4.7	Check that the ticket information is visible on the profile page

name of ticket must be alphanumeric/ cannot have leading/trailing spaces	R5.1	check if update actions are successful only if the ticket names do not contain special characters and do not contain leading/trailing spaces
maximum 60 characters in name	R5.2	check if update actions are successful only if the updated name of the ticket is 60 characters or less and not empty, unsuccessful otherwise
ticket qty must be greater than 1 and less than 100	R5.3	check if update actions are successful only if the updated quantity is between 1 and 100, and unsuccessful otherwise
ticket price must be in the range of [10, 100]	R5.4	check if update actions are successful if the new price is between $10 \leq \text{new price} \leq 100$
Date of ticket must be in YYYYMMDD format	R5.5	check if update is successful if the ticket date is in the correct format
ticket must exist in database	R5.6	check if update is successful when the original/previous ticket name exists in the database
name of ticket must be alphanumeric/ cannot have leading/trailing spaces	R6.1	checks if purchase/buy actions are successful when the entered ticket name is alphanumeric/contains no special characters/leading/trailing spaces
maximum 60 characters in name	R6.2	checks if purchase/buy actions are successful when the entered ticket name is 60 characters or less, and is not empty

ticket qty must be greater than 0 and less/equal than 100	R6.3	checks if purchase/buy actions are successful when the purchase amount is at least 1 and at most 100
ticket name must exist in database and that ticket quantity is more than buy quantity	R6.4	checks if the purchase/buy actions are successful when the ticket being purchased exists in the database and that the total stock/quantity is greater or equal to the purchasing quantity
user has more balance than ticket price * quantity + service fee + tax	R6.5	checks if the purchase/buy actions are successful when the user has enough balance for the selected tickets (i.e. balance > price*qty + service + tax)

Test Cases Allocation:

Joel: R4.1.1, R4.1.2, R4.1.3, R4.1.4, R4.2.1, R4.2.2, R4.3.1

Johnson: R4.3.2, R4.3.3, R4.4.1, R4.4.2, R4.4.3, R4.5.1

Isaac: R4.5.2, R4.6, R4.7, R5.1, R5.2, R5.3, R5.4

Xin Yang: R5.5, R5.6, R6.1, R6.2, R6.3, R6.4, R6.5

Template Modifications

The template that was given by the professor was modified. The changes are documented in the failure report below.

Failure Report

Test Name	What Was Being Tested	How The Output Was Wrong	Error	Fix
R4.2.1	Checking that the selling actions succeeded when the ticket name is < 60 characters	Couldn't find "ticket posted for sale" message on page	Couldn't sell ticket because ticket with same name had been posted in earlier test	Changed ticket name to something unique
R4.3.1	Checking if the selling actions succeeded when the ticket quantity was > 0 and < 101	Couldn't find "ticket posted for sale" message on page	Couldn't sell ticket because ticket with same name had been posted in earlier test	Changed ticket name to something unique
R4.5.2	Sell button	Could not find sell button	Sell button was missing ID	Added id to sell button in index.html
R5.2b	Checking that the empty string "" for name redirect with specific error message	Test did not pass	validate_ticket_name continued to check first and last character when it did not exist	Added condition to prevent checking first and last if the string is empty
R5.3b	Checking that negative update quantity shows correct error message	Test did not pass	Html form requirements (min, max, etc) prevented form from being submitted and the test to continue	Removed said requirements
R6.3e	Checking that purchasing 100 tickets passes checks	Page displayed "Insufficient Funds" instead of "Ticket Purchased"	The mock user did not have sufficient funds for the test case after factoring in tax + service fee	Made new mock user with more funds in balance
R6.4b	Checking that purchasing 4	Page displayed "Insufficient	Mock user did not have sufficient funds	Made new mock user with more

	tickets passes checks	Funds” instead of “Ticket Purchased”	for test case	funds in balance
R6.5b	Check that user cannot purchase ticket due to insufficient balance	Page displayed “Invalid purchase quantity” instead of “Insufficient Funds”	Mock ticket did not have enough quantity to be purchased again	Made new mock ticket
R8.2	Check that valid requests did not bring you to 404 error	Did not check if /buy, /sell, and /update were valid requests	Code was commented out before because r4,r5, and r6 routes were not ready.	Added lines to check if valid requests pointed to their respective routes.

Backend Unit Test

White box testing using statement coverage:

Validate pass

```

1 def validate_password(password):
2     """
3     Validate that password is valid
4     :param password: password to be validated
5     :return: boolean, valid or not
6     """
7     # min length 6
8     if len(password) < 6:
9         return False
10    # at least one upper and one lower
11    if password.isupper() or password.islower():
12        return False
13    # any special character
14    return any(c for c in password if not c.isalnum() and not c.isspace())

```

Stmt	Password	Test
1	“”	T1
2	“”	
3	“”	
4	“”	
5	“”	
6	“”	
7	“”	
8	“”	
9	“”	
10	“”	
11	“AAAAAAAA”	T2
12	“AAAAAAAA”	
13	“AAAAAAAA”	
14	“AbCd12\$”	T3