Aarti, Amaar, Buchi and Lia

# Failure Report Sell,Update,Buy

**Important modifications made to the template:**

- Whole backend was implemented during this stage, the following functions were added:
  - add_ticket(name,quantity,price,expiration)
  - get_ticket(name)
  - remove_ticket(name)
  - update_ticket(name,quantity,price,expiration)
  - update_quantity(name,quantity)
  - get_all_tickets()
- All frontend requirements were not working as intendedly because the names in the forms did not match the python code under the sell,buy and update routes.
- New ticket class was added in models.py that stores the name, price, quantity and expiration date of each ticket being sold.
- Tickets div h4 modified in index.html to display the entire ticket information properly.
  - `<h4>Title: {{ ticket.name }}, Price: {{ ticket.price }}, Quantity: {{ ticket.quantity}}, Expiration Date: {{ ticket.expirationdate}} </h4>`
- Buy,sell and update post routes edited to perform the correct functions if forms are filled correctly i.e. no error message.
  - Buy_post calls backend methods remove_ticket and update_quantity when ticket is bought, updates user balance
  - Update_post calls backend method update_ticket with the inputs in the forms fields
  - Sell_post calls backend methods add_ticket and get_all_tickets

| Test name | What | Output | Error in code | Changes made |
|---|---|---|---|---|
| **All tests passed once the above changes were made and the website worked as required.** | | | | |

**General modifications to test cases:**

- Mocking was used to ensure faster tests and avoid test user to repeatedly register
- Lines with : self.click('input[type="submit"]') were changed. Initially, the tests would self fill the update form but click the submit button for the buy form and hence call the buy_post method. The tests were not calling the appropriate routes. These lines were changed to click the specific button of that form using the elements id. For example, for the buy form, the line `self.click("#sell-btn-submit")` was used.
- The buy was not working because the user balance was not instantiated upon making the test user. Added the user balance into the test user mock.
- Buy was not working because in the front end, we checked whether the ticket existed first, so it did not check the other cases for the invalid tests because the ticket did not exist so it would always show the ticket does not exist.

**Documentation on the backend unit test case creation (e.g. input partitions and test cases for each partition):**

- Backend unit test: get_ticket(name)
- If get_ticket finds a ticket that exists, return the ticket
- If get_ticket does not find a ticket that exists, return None
- Statement analysis/coverage was used

**Integration test cases passed: user walkthroughs for creating and purchasing tickets were made in two different files.**