

# A6: Integration and System Delivery

Group 25

## A6: Integration and System Delivery

Patrick Gaskin (20217404) Qays Ebrahim (20294208) Jeff Li (20221839)

### Test Script Listing

| Script             | Description   | Type              |
|--------------------|---|-------------------|
| a5_test_acctupd.py | White-box statement coverage tests for the account update form POST method. Doesn't include the validation itself (we already tested that in the black-box tests), only that we actually use the validation result and show a relevant error message when processing an account update. | Custom            |
| a5_test_ost.py     | White-box data interface coverage tests for order state transitions.  | Custom            |
| validate_test.py   | Black-box tests for the validation functions using input partitioning.  | PyTest            |
| main_test.py       | Requirements and integration tests.   | Custom + Selenium |

### Instructions

1. Ensure you have Python 3.11+ installed. On Windows, it may be called “py” rather than “python3” depending on how you installed it.
2. Optionally create a venv using `python3 -m venv .env`, then activate it.
3. Install the dependencies using `python3 -m pip install -r requirements.txt`.
4. Optionally run `python3 a5_test_acctupd.py`. The test results will be written to the console.
5. Optionally run `python3 a5_test_ost.py`. The test results will be written to the console.
6. Optionally run `pytest -rA` (for `validate_test.py`). The test results will be written to the console.
7. Optionally run `python3 main_test.py`. The test results will be written to the console and `A4/A4_Part1.md`, and the screenshots will be written to `A4/`.
8. Run `python3 main.py` to start the application on `localhost:8081`.

### Output

See next page.

## main\_test.py

See main\_test.pdf for the screenshots.

```
$ python3 main_test.py
... loading test cases
... starting chrome

=== EXECUTING 1.1 Create account (Invalid password)
... starting restaurant server
--- InitialState
... logging in as None with password 'password'
... navigating to /account/create
... saving screenshot
--- InputData
... setting 'username' to 'john'
... setting 'password' to ''
... setting 'first_name' to 'John'
... setting 'last_name' to 'Smith'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Invalid account information')
??? running check Page('/account/create')
*** TEST PASSED

=== EXECUTING 1.2 Create account (Invalid username)
... starting restaurant server
--- InitialState
... logging in as None with password 'password'
... navigating to /account/create
... saving screenshot
--- InputData
... setting 'username' to 'john!'
... setting 'password' to 'password'
... setting 'first_name' to 'John'
... setting 'last_name' to 'Smith'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Invalid account information')
??? running check Page('/account/create')
*** TEST PASSED

=== EXECUTING 1.3 Create account (Invalid username)
... starting restaurant server
--- InitialState
... logging in as None with password 'password'
... navigating to /account/create
... saving screenshot
```

```

--- InputData
... setting 'username' to 'john smith'
... setting 'password' to 'password'
... setting 'first_name' to 'John'
... setting 'last_name' to 'Smith'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Invalid account information')
??? running check Page('/account/create')
*** TEST PASSED

```

=== EXECUTING 1.4 Create account (Duplicate username)

```

... starting restaurant server
--- InitialState
... logging in as None with password 'password'
... navigating to /account/create
... saving screenshot
--- InputData
... setting 'username' to 'patrick'
... setting 'password' to 'password'
... setting 'first_name' to 'Patrick'
... setting 'last_name' to 'Smith'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Username already exists')
??? running check Page('/account/create')
*** TEST PASSED

```

=== EXECUTING 1.5 Create account (Valid)

```

... starting restaurant server
--- InitialState
... logging in as None with password 'password'
... navigating to /account/create
... saving screenshot
--- InputData
... setting 'username' to 'john'
... setting 'password' to 'password'
... setting 'first_name' to 'John'
... setting 'last_name' to 'Smith'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Account created')
??? running check Page('/account')
*** TEST PASSED

```

```

=== EXECUTING 2.1 Modify account (Valid password change)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /account
... saving screenshot
--- InputData
... setting 'firstname' to 'Patrick'
... setting 'lastname' to 'Gaskin'
... setting 'password' to 'password1'
... setting 'address' to '123 Main St, Kingston, Ontario, Canada A1B 2C3'
... setting 'cardnumber' to '2222420000001113'
... setting 'cardexpiry' to '08/26'
... setting 'cardcode' to '123'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Account updated')
??? running check Page('/account')
*** TEST PASSED

=== EXECUTING 2.2 Modify account (Invalid card information change)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /account
... saving screenshot
--- InputData
... setting 'firstname' to 'Patrick'
... setting 'lastname' to 'Gaskin'
... setting 'password' to ''
... setting 'address' to '123 Main St, Kingston, Ontario, Canada A1B 2C3'
... setting 'cardnumber' to '1'
... setting 'cardexpiry' to '99/99'
... setting 'cardcode' to '123'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Invalid card information')
??? running check Page('/account')
*** TEST PASSED

=== EXECUTING 2.3 Modify account (Valid card information and password change)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /account
... saving screenshot

```

```

--- InputData
... setting 'firstname' to 'Patrick'
... setting 'lastname' to 'Gaskin'
... setting 'password' to 'password1'
... setting 'address' to '123 Main St, Kingston, Ontario, Canada A1B 2C3'
... setting 'cardnumber' to '2223000048410010'
... setting 'cardexpiry' to '09/26'
... setting 'cardcode' to '123'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Account updated')
??? running check Page('/account')
*** TEST PASSED

```

```

=== EXECUTING 3.1 Log in (Incorrect username)
... starting restaurant server
--- InitialState
... logging in as None with password 'password'
... navigating to /account/login
... saving screenshot
--- InputData
... setting 'username' to 'user'
... setting 'password' to 'password'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('User does not exist')
??? running check Page('/account/login')
*** TEST PASSED

```

```

=== EXECUTING 3.2 Log in (Incorrect password)
... starting restaurant server
--- InitialState
... logging in as None with password 'password'
... navigating to /account/login
... saving screenshot
--- InputData
... setting 'username' to 'patrick'
... setting 'password' to 'dummy'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Incorrect password')
??? running check Page('/account/login')
*** TEST PASSED

```

```

=== EXECUTING 3.3 Log in (Success)
... starting restaurant server
--- InitialState
... logging in as None with password 'password'
... navigating to /account/login
... saving screenshot
--- InputData
... setting 'username' to 'patrick'
... setting 'password' to 'password'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Page('/restaurants')
??? running check LoggedInUser('patrick')
*** TEST PASSED

=== EXECUTING 4.1 Log out (Success)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /account/logout
... saving screenshot
--- ExpectedOutput
... saving screenshot
??? running check LoggedInUser(None)
??? running check Page('/restaurants')
*** TEST PASSED

=== EXECUTING 5.1 List customer orders (List customer orders)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /orders
... saving screenshot
--- ExpectedOutput
... saving screenshot
??? running check VisibleOrderIDs([2])
*** TEST PASSED

=== EXECUTING 6.1 List restaurants (Display list of restaurants)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants
... saving screenshot
--- ExpectedOutput
... saving screenshot
??? running check VisibleRestaurantIDs([1, 2])
*** TEST PASSED

=== EXECUTING 7.1 View restaurant (Customer+Owner)

```

```

... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- ExpectedOutput
... saving screenshot
??? running check RestaurantName('My Restaurant')
??? running check VisibleMenuItems(['Tea', 'Pop', 'Water'])
??? running check VisibleButtons(['restaurant:order', 'restaurant:delete', 'restaurant:update', 'employee:new:add', 'employee:qays:rem
*** TEST PASSED

=== EXECUTING 7.2 View restaurant (Customer+Employee)
... starting restaurant server
--- InitialState
... logging in as qays with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- ExpectedOutput
... saving screenshot
??? running check RestaurantName('My Restaurant')
??? running check VisibleMenuItems(['Tea', 'Pop', 'Water'])
??? running check VisibleButtons(['restaurant:order', 'order:2:accept', 'order:1:deliver'])
??? running check VisibleOrderIDs([1, 2])
*** TEST PASSED

=== EXECUTING 7.3 View restaurant (Customer)
... starting restaurant server
--- InitialState
... logging in as jeff with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- ExpectedOutput
... saving screenshot
??? running check RestaurantName('My Restaurant')
??? running check VisibleMenuItems(['Tea', 'Pop', 'Water'])
??? running check VisibleButtons(['restaurant:order'])
*** TEST PASSED

=== EXECUTING 8.1 New order (Valid new order)
... starting restaurant server
--- InitialState
... logging in as jeff with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... using action 'restaurant:order'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Page('/orders/4')
??? running check VisibleButtons(['order:pay', 'order:cancel'])

```

\*\*\* TEST PASSED

=== EXECUTING 9.1 Add/remove item in order (Add)

... starting restaurant server

--- InitialState

... logging in as qays with password 'password'

... navigating to /orders/3

... saving screenshot

--- InputData

... using action 'item:4:add'

... saving screenshot

--- ExpectedOutput

... clicking action

... saving screenshot

??? running check OrderItems(['Tea=1', 'Pop=1', 'Water=0'])

??? running check Page('/orders/3')

\*\*\* TEST PASSED

=== EXECUTING 9.2 Add/remove item in order (Remove)

... starting restaurant server

--- InitialState

... logging in as qays with password 'password'

... navigating to /orders/3

... saving screenshot

--- InputData

... using action 'item:2:subtract'

... saving screenshot

--- ExpectedOutput

... clicking action

... saving screenshot

??? running check OrderItems(['Tea=0', 'Pop=0', 'Water=0'])

??? running check Page('/orders/3')

\*\*\* TEST PASSED

=== EXECUTING 9.3 Add/remove item in order (Invalid)

... starting restaurant server

--- InitialState

... logging in as qays with password 'password'

... navigating to /orders/3

... saving screenshot

--- InputData

... using FORCED (i.e., not actually there) action 'item:6:add'

... saving screenshot

--- ExpectedOutput

... clicking action

... saving screenshot

??? running check OrderItems(['Tea=1', 'Pop=0', 'Water=0'])

??? running check Page('/orders/3')

\*\*\* TEST PASSED

=== EXECUTING 10.1 Submit order (Valid)

... starting restaurant server

--- InitialState



```

... logging in as qays with password 'password'
... navigating to /orders/3
... saving screenshot
--- InputData
... using action 'order:pay'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Order submitted')
??? running check Page('/orders/3')
??? running check VisibleButtons(['order:cancel'])
*** TEST PASSED

=== EXECUTING 11.1 Cancel order (Invalid)
... starting restaurant server
--- InitialState
... logging in as jeff with password 'password'
... navigating to /orders/1
... saving screenshot
--- InputData
... using FORCED (i.e., not actually there) action 'order:cancel'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Bad action')
??? running check Page('/orders/1')
??? running check OrderStatus('ACCEPTED')
*** TEST PASSED

=== EXECUTING 11.2 Cancel order (Valid)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /orders/2
... saving screenshot
--- InputData
... using action 'order:cancel'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Order cancelled')
??? running check Page('/orders/2')
??? running check OrderStatus('CANCELLED')
*** TEST PASSED

=== EXECUTING 11.3 Cancel order (Valid)
... starting restaurant server
--- InitialState
... logging in as qays with password 'password'
... navigating to /orders/3

```

```

... saving screenshot
--- InputData
... using action 'order:cancel'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Order cancelled')
??? running check Page('/orders/3')
??? running check OrderStatus('CANCELLED')
*** TEST PASSED

=== EXECUTING 12.1 Create restaurant (Invalid)
... starting restaurant server
--- InitialState
... logging in as jeff with password 'password'
... navigating to /restaurants
... saving screenshot
--- InputData
... setting 'name' to ''
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('must not be blank')
??? running check Page('/restaurants')
*** TEST PASSED

=== EXECUTING 12.2 Create restaurant (Valid)
... starting restaurant server
--- InitialState
... logging in as jeff with password 'password'
... navigating to /restaurants
... saving screenshot
--- InputData
... setting 'name' to 'Test'
... using submit button
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Restaurant created')
??? running check Page('/restaurants')
??? running check RestaurantListContainsName(['Test'])
*** TEST PASSED

=== EXECUTING 13.1 Update restaurant (Success)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot

```

```

--- InputData
... setting 'restaurant:name' to 'Best Restaurant'
... using action 'restaurant:update'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check RestaurantName('Best Restaurant')
??? running check Page('/restaurants/1')
*** TEST PASSED

=== EXECUTING 14.1 Delete restaurant (Success)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... using action 'restaurant:delete'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Page('/restaurants')
??? running check NotVisibleRestaurantNames(['My Restaurant'])
*** TEST PASSED

=== EXECUTING 15.1 Add employee (Invalid)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... setting 'employee:new:username' to 'sdfsdf'
... using action 'employee:new:add'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('user does not exist')
??? running check Page('/restaurants/1')
??? running check VisibleEmployeeUsernames(['qays'])
*** TEST PASSED

=== EXECUTING 15.2 Add employee (Valid)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... setting 'employee:new:username' to 'jeff'

```

```

... using action 'employee:new:add'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Page('/restaurants/1')
??? running check VisibleEmployeeUsernames(['qays', 'jeff'])
*** TEST PASSED

=== EXECUTING 16.1 Remove employee (Success)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... using action 'employee:qays:remove'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Page('/restaurants/1')
??? running check VisibleEmployeeUsernames([])
*** TEST PASSED

=== EXECUTING 17.1 Add menu item (Valid)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... setting 'item:new:price' to '3.75'
... setting 'item:new:name' to 'Smoothie'
... using action 'item:new:add'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Page('/restaurants/1')
??? running check VisibleMenuItems(['Tea', 'Pop', 'Water', 'Smoothie'])
*** TEST PASSED

=== EXECUTING 17.2 Add menu item (Invalid)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... setting 'item:new:price' to '-3.75'
... setting 'item:new:name' to 'dummy'
... using action 'item:new:add'

```

```

... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('invalid price')
??? running check Page('/restaurants/1')
??? running check VisibleMenuItems(['Tea', 'Pop', 'Water'])
*** TEST PASSED

=== EXECUTING 17.3 Add menu item (Invalid)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... setting 'item:new:price' to 'one dollar'
... setting 'item:new:name' to 'dummy'
... using action 'item:new:add'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('invalid price')

!!! CHECK FAILED: incorrect message 'could not convert string to float: &#39;one dollar&#39;'
!!! TEST FAILED

=== EXECUTING 17.4 Add menu item (Invalid)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... setting 'item:new:price' to '1.0'
... setting 'item:new:name' to ''
... using action 'item:new:add'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('item name')
??? running check Page('/restaurants/1')
??? running check VisibleMenuItems(['Tea', 'Pop', 'Water'])
*** TEST PASSED

=== EXECUTING 18.1 Update menu item (Valid)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot

```

```

--- InputData
... setting 'item:2:name' to 'Tea (special)'
... setting 'item:2:price' to '0.5'
... using action 'item:2:update'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check VisibleMenuItems(['Tea (special)', 'Pop', 'Water'])
??? running check Page('/restaurants/1')
*** TEST PASSED

```

```

=== EXECUTING 18.2 Update menu item (Invalid)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... setting 'item:2:name' to ''
... setting 'item:2:price' to '1.2'
... using action 'item:2:update'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('item name must not be empty')
??? running check VisibleMenuItems(['Tea', 'Pop', 'Water'])
??? running check Page('/restaurants/1')
*** TEST PASSED

```

```

=== EXECUTING 19.1 Delete menu item (Valid)
... starting restaurant server
--- InitialState
... logging in as patrick with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... using action 'item:2:delete'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Page('/restaurants/1')
??? running check VisibleMenuItems(['Pop', 'Water'])
*** TEST PASSED

```

```

=== EXECUTING 20.1 Accept restaurant order (Success)
... starting restaurant server
--- InitialState
... logging in as qays with password 'password'
... navigating to /restaurants/1
... saving screenshot

```

```

--- InputData
... using action 'order:2:accept'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Order accepted')
??? running check VisibleButtons(['order:2:deliver'])
??? running check Page('/restaurants/1')
*** TEST PASSED

=== EXECUTING 21.1 Deliver restaurant order (Success)
... starting restaurant server
--- InitialState
... logging in as qays with password 'password'
... navigating to /restaurants/1
... saving screenshot
--- InputData
... using action 'order:1:deliver'
... saving screenshot
--- ExpectedOutput
... clicking action
... saving screenshot
??? running check Message('Order delivered')
??? running check NotVisibleOrderIDs([1])
??? running check Page('/restaurants/1')
*** TEST PASSED

```

a5\_test\_ost.py

\$ python3 a5\_test\_ost.py

```
> RUN pending order: invalid order id for restaurant
[N] PENDING -> ACCEPTED :: no such order 100 for restaurant 1
[N] PENDING -> CANCELLED :: no such order 100 for restaurant 1
[N] PENDING -> DELIVERED :: no such order 100 for restaurant 1
[N] PENDING -> PAID :: no such order 100 for restaurant 1
- PASS allowed transitions none

> RUN pending order: restaurant is deleted
[N] PENDING -> ACCEPTED :: restaurant does not exist or has been deleted
[N] PENDING -> CANCELLED :: restaurant does not exist or has been deleted
[N] PENDING -> DELIVERED :: restaurant does not exist or has been deleted
[N] PENDING -> PAID :: restaurant does not exist or has been deleted
- PASS allowed transitions none

> RUN pending order: order was created by a different customer
[N] PENDING -> ACCEPTED :: bad transition PENDING -> ACCEPTED
[N] PENDING -> CANCELLED :: cannot cancel someone else's order
[N] PENDING -> DELIVERED :: bad transition PENDING -> DELIVERED
[N] PENDING -> PAID :: cannot pay for someone else's order
- PASS allowed transitions none

> RUN pending order: customer does not have billing information
[N] PENDING -> ACCEPTED :: bad transition PENDING -> ACCEPTED
[Y] PENDING -> CANCELLED
[N] PENDING -> DELIVERED :: bad transition PENDING -> DELIVERED
[N] PENDING -> PAID :: cannot pay for order without address and billing information set for account
- PASS allowed transitions PENDING->CANCELLED

> RUN pending order: order contains items it shouldn't
[N] PENDING -> ACCEPTED :: bad transition PENDING -> ACCEPTED
[Y] PENDING -> CANCELLED
[N] PENDING -> DELIVERED :: bad transition PENDING -> DELIVERED
[N] PENDING -> PAID :: order contains item Burger from another restaurant (wtf... are you messing with the requests or the
- PASS allowed transitions PENDING->CANCELLED

> RUN pending order: order contains deleted item
[N] PENDING -> ACCEPTED :: bad transition PENDING -> ACCEPTED
[Y] PENDING -> CANCELLED
[N] PENDING -> DELIVERED :: bad transition PENDING -> DELIVERED
[N] PENDING -> PAID :: cannot order deleted item Burger
- PASS allowed transitions PENDING->CANCELLED

> RUN pending order: order does not contain any items
[N] PENDING -> ACCEPTED :: bad transition PENDING -> ACCEPTED
[Y] PENDING -> CANCELLED
[N] PENDING -> DELIVERED :: bad transition PENDING -> DELIVERED
[N] PENDING -> PAID :: order must contain at least one item
- PASS allowed transitions PENDING->CANCELLED

> RUN is someone else's order (not an employee)
```



```

[N] PAID      -> PENDING    :: bad transition PAID -> PENDING
[N] PAID      -> ACCEPTED   :: cannot accept an order as a non-employee
[N] PAID      -> CANCELLED  :: cannot cancel someone else's order
[N] PAID      -> DELIVERED  :: bad transition PAID -> DELIVERED
[N] CANCELLED -> PENDING    :: bad transition CANCELLED -> PENDING
[N] CANCELLED -> ACCEPTED   :: bad transition CANCELLED -> ACCEPTED
[N] CANCELLED -> DELIVERED  :: bad transition CANCELLED -> DELIVERED
[N] CANCELLED -> PAID       :: bad transition CANCELLED -> PAID
[N] ACCEPTED  -> PENDING    :: bad transition ACCEPTED -> PENDING
[N] ACCEPTED  -> CANCELLED  :: accepted order cannot be cancelled
[N] ACCEPTED  -> DELIVERED  :: cannot deliver an order as a non-employee
[N] ACCEPTED  -> PAID       :: bad transition ACCEPTED -> PAID
[N] PENDING   -> ACCEPTED   :: bad transition PENDING -> ACCEPTED
[N] PENDING   -> CANCELLED  :: cannot cancel someone else's order
[N] PENDING   -> DELIVERED  :: bad transition PENDING -> DELIVERED
[N] PENDING   -> PAID       :: cannot pay for someone else's order
[N] DELIVERED -> PENDING    :: bad transition DELIVERED -> PENDING
[N] DELIVERED -> ACCEPTED   :: bad transition DELIVERED -> ACCEPTED
[N] DELIVERED -> CANCELLED  :: bad transition DELIVERED -> CANCELLED
[N] DELIVERED -> PAID       :: bad transition DELIVERED -> PAID

```

- PASS allowed transitions none

> RUN is user's order (not an employee)

```

[N] PAID      -> PENDING    :: bad transition PAID -> PENDING
[N] PAID      -> ACCEPTED   :: cannot accept an order as a non-employee
[Y] PAID      -> CANCELLED
[N] PAID      -> DELIVERED  :: bad transition PAID -> DELIVERED
[N] CANCELLED -> PENDING    :: bad transition CANCELLED -> PENDING
[N] CANCELLED -> ACCEPTED   :: bad transition CANCELLED -> ACCEPTED
[N] CANCELLED -> DELIVERED  :: bad transition CANCELLED -> DELIVERED
[N] CANCELLED -> PAID       :: bad transition CANCELLED -> PAID
[N] ACCEPTED  -> PENDING    :: bad transition ACCEPTED -> PENDING
[N] ACCEPTED  -> CANCELLED  :: accepted order cannot be cancelled
[N] ACCEPTED  -> DELIVERED  :: cannot deliver an order as a non-employee
[N] ACCEPTED  -> PAID       :: bad transition ACCEPTED -> PAID
[N] PENDING   -> ACCEPTED   :: bad transition PENDING -> ACCEPTED
[Y] PENDING   -> CANCELLED
[N] PENDING   -> DELIVERED  :: bad transition PENDING -> DELIVERED
[Y] PENDING   -> PAID
[N] DELIVERED -> PENDING    :: bad transition DELIVERED -> PENDING
[N] DELIVERED -> ACCEPTED   :: bad transition DELIVERED -> ACCEPTED
[N] DELIVERED -> CANCELLED  :: bad transition DELIVERED -> CANCELLED
[N] DELIVERED -> PAID       :: bad transition DELIVERED -> PAID

```

- PASS allowed transitions PENDING->CANCELLED, PENDING->PAID, PAID->CANCELLED

> RUN is someone else's order (is an employee)

```

[N] PAID      -> PENDING    :: bad transition PAID -> PENDING
[Y] PAID      -> ACCEPTED
[N] PAID      -> CANCELLED  :: cannot cancel someone else's order
[N] PAID      -> DELIVERED  :: bad transition PAID -> DELIVERED
[N] CANCELLED -> PENDING    :: bad transition CANCELLED -> PENDING
[N] CANCELLED -> ACCEPTED   :: bad transition CANCELLED -> ACCEPTED
[N] CANCELLED -> DELIVERED  :: bad transition CANCELLED -> DELIVERED

```

```

[N] CANCELLED -> PAID      :: bad transition CANCELLED -> PAID
[N] ACCEPTED  -> PENDING   :: bad transition ACCEPTED -> PENDING
[N] ACCEPTED  -> CANCELLED :: accepted order cannot be cancelled
[Y] ACCEPTED  -> DELIVERED
[N] ACCEPTED  -> PAID      :: bad transition ACCEPTED -> PAID
[N] PENDING   -> ACCEPTED  :: bad transition PENDING -> ACCEPTED
[N] PENDING   -> CANCELLED :: cannot cancel someone else's order
[N] PENDING   -> DELIVERED :: bad transition PENDING -> DELIVERED
[N] PENDING   -> PAID      :: cannot pay for someone else's order
[N] DELIVERED -> PENDING   :: bad transition DELIVERED -> PENDING
[N] DELIVERED -> ACCEPTED  :: bad transition DELIVERED -> ACCEPTED
[N] DELIVERED -> CANCELLED :: bad transition DELIVERED -> CANCELLED
[N] DELIVERED -> PAID      :: bad transition DELIVERED -> PAID
- PASS allowed transitions PAID->ACCEPTED, ACCEPTED->DELIVERED

> RUN is user's order (is an employee)
[N] PAID      -> PENDING   :: bad transition PAID -> PENDING
[Y] PAID      -> ACCEPTED
[Y] PAID      -> CANCELLED
[N] PAID      -> DELIVERED :: bad transition PAID -> DELIVERED
[N] CANCELLED -> PENDING   :: bad transition CANCELLED -> PENDING
[N] CANCELLED -> ACCEPTED  :: bad transition CANCELLED -> ACCEPTED
[N] CANCELLED -> DELIVERED :: bad transition CANCELLED -> DELIVERED
[N] CANCELLED -> PAID      :: bad transition CANCELLED -> PAID
[N] ACCEPTED  -> PENDING   :: bad transition ACCEPTED -> PENDING
[N] ACCEPTED  -> CANCELLED :: accepted order cannot be cancelled
[Y] ACCEPTED  -> DELIVERED
[N] ACCEPTED  -> PAID      :: bad transition ACCEPTED -> PAID
[N] PENDING   -> ACCEPTED  :: bad transition PENDING -> ACCEPTED
[Y] PENDING   -> CANCELLED
[N] PENDING   -> DELIVERED :: bad transition PENDING -> DELIVERED
[Y] PENDING   -> PAID
[N] DELIVERED -> PENDING   :: bad transition DELIVERED -> PENDING
[N] DELIVERED -> ACCEPTED  :: bad transition DELIVERED -> ACCEPTED
[N] DELIVERED -> CANCELLED :: bad transition DELIVERED -> CANCELLED
[N] DELIVERED -> PAID      :: bad transition DELIVERED -> PAID
- PASS allowed transitions PAID->ACCEPTED, PAID->CANCELLED, PENDING->PAID, PENDING->CANCELLED, ACCEPTED->DELIVERED

all tests passed

```

a5\_test\_acctupd.py

```
$ python3 a5_test_acctupd.py
```

```
starting chrome
```

```
starting restaurant server
```

```
logging in as jeff:password
```

```
attempting account update with fields [('firstname', 'Test'), ('lastname', 'Test'), ('password', 'password1'), ('address', '123 Main S
```

```
... got error message: Account updated.
```

```
attempting account update with fields [('firstname', 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
... got error message: Invalid first or last name (first name too long).
```

```
attempting account update with fields [('lastname', 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
... got error message: Invalid first or last name (first name too long).
```

```
attempting account update with fields [('password', 'short')]
```

```
... got error message: Invalid password (password must be at least 6 characters long).
```

```
attempting account update with fields [('address', 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
... got error message: Invalid address (address too long).
```

```
attempting account update with fields [('cardnumber', 'abc')]
```

```
... got error message: Invalid card information (card number must only contain numbers).
```

```
attempting account update with fields [('cardexpiry', '99/99')]
```

```
... got error message: Invalid card information (invalid card expiry month).
```

```
attempting account update with fields [('cardcode', 'asfgds')]
```

```
... got error message: Invalid card information (invalid card code).
```

```
attempting account update with fields []
```

```
... clearing cookies to simulate submitting account information form without being logged in
```

```
... got error message: Not logged in.
```

```
all tests passed
```

validate\_test.py

\$ pytest -rA

===== test session starts =====

platform linux -- Python 3.12.0, pytest-7.4.3, pluggy-1.3.0

rootdir: /home/patrick/qu/2324/CMPE327/CISC327\_F23

collected 85 items

validate\_test.py .....

===== PASSES =====

===== short test summary info =====

PASSED validate\_test.py::test\_validate\_account\_username[-None]  
PASSED validate\_test.py::test\_validate\_account\_username[ -None]  
PASSED validate\_test.py::test\_validate\_account\_username[ -None]  
PASSED validate\_test.py::test\_validate\_account\_username[abcdefghijklnopqrstuvwxyz-None]  
PASSED validate\_test.py::test\_validate\_account\_username[ abcdefghijklnopqrstuvwxyz -None]  
PASSED validate\_test.py::test\_validate\_account\_username[abc123#@-None]  
PASSED validate\_test.py::test\_validate\_account\_username[\\xf6\\xe4\\xfc\\xdf-None]  
PASSED validate\_test.py::test\_validate\_account\_username[abCdef-None]  
PASSED validate\_test.py::test\_validate\_account\_username[abc-def-None]  
PASSED validate\_test.py::test\_validate\_account\_username[abc def-None]  
PASSED validate\_test.py::test\_validate\_account\_username[abcdefghijklnopqrstuvwxyz-abcdefghijklnopqrstuvwxyz]  
PASSED validate\_test.py::test\_validate\_account\_username[ab\_c-ab\_c]  
PASSED validate\_test.py::test\_validate\_account\_username[ab.c-ab.c]  
PASSED validate\_test.py::test\_validate\_account\_username[ dabcdeffhijklmn-dabcdeffhijklmn]  
PASSED validate\_test.py::test\_validate\_account\_username[dabcdeffhijklmn-dabcdeffhijklmn]  
PASSED validate\_test.py::test\_validate\_account\_password[jhs6-None]  
PASSED validate\_test.py::test\_validate\_account\_password[-None]  
PASSED validate\_test.py::test\_validate\_account\_password[ -None]  
PASSED validate\_test.py::test\_validate\_account\_password[ahs73g-ahs73g]  
PASSED validate\_test.py::test\_validate\_account\_password[ahs73gD-ahs73gD]  
PASSED validate\_test.py::test\_validate\_account\_password[ABC12@3DEF-ABC12@3DEF]  
PASSED validate\_test.py::test\_validate\_restaurant\_name[-None]  
PASSED validate\_test.py::test\_validate\_restaurant\_name[ -None]  
PASSED validate\_test.py::test\_validate\_restaurant\_name[ -None]  
PASSED validate\_test.py::test\_validate\_restaurant\_name[dD5PmKQ7SU5ornjA0v5J7VrzNtYqRgmo0fksFYvrRaFjdhsdhjsdhsjdhsTvaEVmJyGDtPtE6sNbgEX  
PASSED validate\_test.py::test\_validate\_restaurant\_name[ dD5PmKQ7SU5ornjA0v5J7VrzNtYqRgmo0fksFYvrRadsdjkdsjksdjskdjskFTvaEVmJyGDtP  
PASSED validate\_test.py::test\_validate\_restaurant\_name[dD5PmKQ7SU5ornjA0v5J7VrzNtYqRgmo0fksFYvrRaFjdhsdhjsdhsjdhsTvaEVmJyGDtPtE6sNbgEX  
PASSED validate\_test.py::test\_validate\_restaurant\_name[241Pizza-241Pizza]  
PASSED validate\_test.py::test\_validate\_restaurant\_name[McDonalds-McDonalds]  
PASSED validate\_test.py::test\_validate\_restaurant\_name[ subway -subway]  
PASSED validate\_test.py::test\_validate\_account\_card\_number[-None]  
PASSED validate\_test.py::test\_validate\_account\_card\_number[ -None]  
PASSED validate\_test.py::test\_validate\_account\_card\_number[ -None]  
PASSED validate\_test.py::test\_validate\_account\_card\_number[123412341234123d-None]  
PASSED validate\_test.py::test\_validate\_account\_card\_number[378282246310005-378282246310005]  
PASSED validate\_test.py::test\_validate\_account\_card\_number[371449635398431-371449635398431]  
PASSED validate\_test.py::test\_validate\_account\_card\_number[378734493671000-378734493671000]  
PASSED validate\_test.py::test\_validate\_account\_card\_code[-None0]  
PASSED validate\_test.py::test\_validate\_account\_card\_code[-None1]  
PASSED validate\_test.py::test\_validate\_account\_card\_code[1212-None]  
PASSED validate\_test.py::test\_validate\_account\_card\_code[L21-None]  
PASSED validate\_test.py::test\_validate\_account\_card\_code[\\u066421-None]

```

PASSED validate_test.py::test_validate_account_card_code[212-212]
PASSED validate_test.py::test_validate_account_card_code[988-988]
PASSED validate_test.py::test_validate_account_card_expiry[-None]
PASSED validate_test.py::test_validate_account_card_expiry[1212-None]
PASSED validate_test.py::test_validate_account_card_expiry[121212-None]
PASSED validate_test.py::test_validate_account_card_expiry[12-12-None]
PASSED validate_test.py::test_validate_account_card_expiry[\u06642156-None]
PASSED validate_test.py::test_validate_account_card_expiry[12/1a-None]
PASSED validate_test.py::test_validate_account_card_expiry[1a/12-None]
PASSED validate_test.py::test_validate_account_card_expiry[12/12-12/12]
PASSED validate_test.py::test_validate_account_card_expiry[01/01-01/01]
PASSED validate_test.py::test_validate_account_address[-None]
PASSED validate_test.py::test_validate_account_address[ -None]
PASSED validate_test.py::test_validate_account_address[      -None]
PASSED validate_test.py::test_validate_account_address[vHH809NUeF8VQUdtORuOYxyMFArYtZh2gWtoo87sJpo0HOftepkx6GYr880HjR7gTX47TfVa7PpD5tA
PASSED validate_test.py::test_validate_account_address[vHH809NUeF8VQUdtORuOYxyMFArYtZh2gWtoo87sJpo0HOftepkx6GYr880HjR7gTX47TfVa7PpD5tA
PASSED validate_test.py::test_validate_account_address[dd5PmKQ7SU5ornjA0v5J7VrzNtYqRgmo0fksFYvrRaFjdhsdhjsdhsjdhsTvaEVmJyGDtPtE6sNbgEX
PASSED validate_test.py::test_validate_account_address[1234 Street AVE-1234 Street AVE]
PASSED validate_test.py::test_validate_account_address[ 1234 Street AVE  -1234 Street AVE]
PASSED validate_test.py::test_validate_menu_item[-None]
PASSED validate_test.py::test_validate_menu_item[KjNzOuRpHsEfWcXqUdKyGwHbAeTiFzEiCmSdJlOrWnLuAfGvZxNpIyHbVuTaLqXjKraPxoLrEhAsDfFkYeLbV
PASSED validate_test.py::test_validate_menu_item[dd5PmKQ7SU5ornjA0v5J7VrzNtYqRgmo0fksFYvrRaFjdhsdhjsdhsjdhsTvaEVmJyGDtPtE6sNbgEXyvWFnY
PASSED validate_test.py::test_validate_menu_item[Pizza-Pizza]
PASSED validate_test.py::test_validate_first_last_name[-None]
PASSED validate_test.py::test_validate_first_last_name[wFzLpHgDxKcYqZoUvEaIbNfRjQsXkHlMmCvGuTtXhJiYnOaEoVrZpSbCpWlPvQaAqGdUcLgXkWwYjXr
PASSED validate_test.py::test_validate_first_last_name[dd5PmKQ7SU5ornjA0v5J7VrzNtYqRgmo0fksFYvrRaFjdhsdhjsdhsjdhsTvaEVmJyGDtPtE6sNbgEX
PASSED validate_test.py::test_validate_first_last_name[John-John]
PASSED validate_test.py::test_validate_price[-None]
PASSED validate_test.py::test_validate_price[$-None]
PASSED validate_test.py::test_validate_price[ $ -None]
PASSED validate_test.py::test_validate_price[a-None]
PASSED validate_test.py::test_validate_price[1.2.3-None]
PASSED validate_test.py::test_validate_price[1x-None]
PASSED validate_test.py::test_validate_price[-1-None]
PASSED validate_test.py::test_validate_price[1.333-None]
PASSED validate_test.py::test_validate_price[1-100]
PASSED validate_test.py::test_validate_price[1.33-133]
PASSED validate_test.py::test_validate_price[1.3-130]
PASSED validate_test.py::test_validate_price[ 1.3-130]
PASSED validate_test.py::test_validate_price[$ 1.3-130]
PASSED validate_test.py::test_validate_price[ $ 1.3-130]
PASSED validate_test.py::test_validate_price[ $ 1.30 -130]
PASSED validate_test.py::test_validate_price[ $ 0001.30000 -130]
===== 85 passed in 0.37s =====

```