main\_test

Group 25

.

# 1. Create account

Objective: Allow user to create an account.

Arrange: Open account creation page.

Act: Fill in user information (username, password).

Assert: If the user doesn't already exist and the password is valid, the user should be created, logged in, and redirected to the account details page.

# 1.1. Create account (Invalid password)

ExpectedOutput:

Message: Invalid account information

Page: /account/create

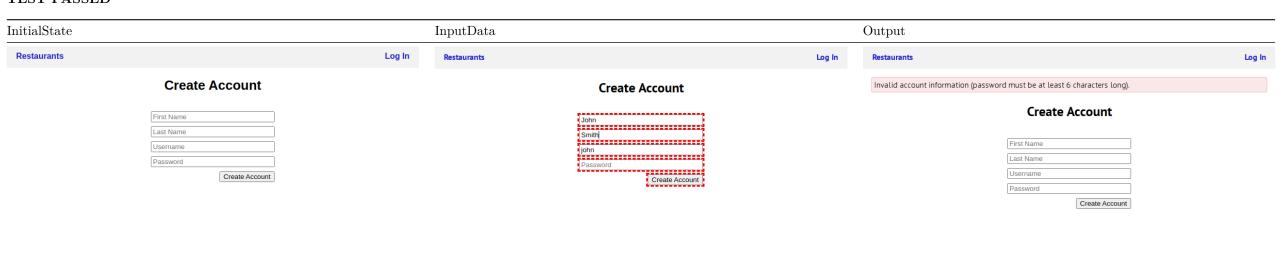
InitialState:

LoggedInUser: null
Page: /account/create

InputData:

first\_name: John
last\_name: Smith
password: ''
username: john

Name: Invalid password



# 1.2. Create account (Invalid username)

ExpectedOutput:

Message: Invalid account information

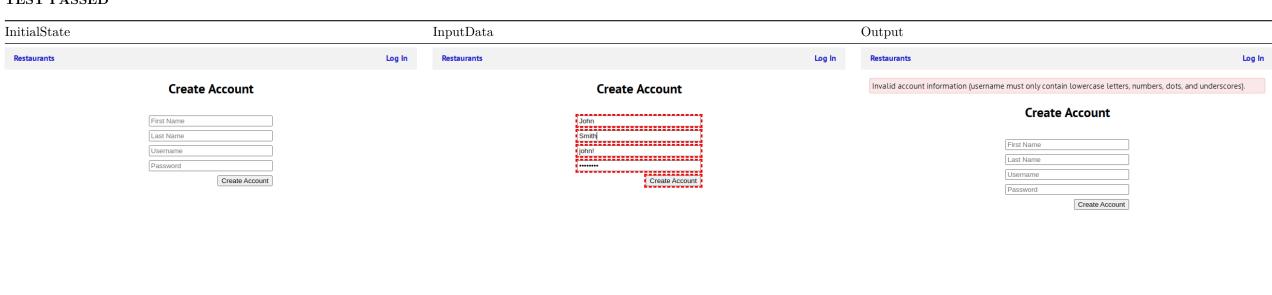
Page: /account/create

InitialState:

LoggedInUser: null
Page: /account/create

InputData:

first\_name: John
last\_name: Smith
password: password
username: john!
Name: Invalid username



# 1.3. Create account (Invalid username)

ExpectedOutput:

Message: Invalid account information

Page: /account/create

InitialState:

LoggedInUser: null
Page: /account/create

InputData:

first\_name: John
last\_name: Smith
password: password
username: john smith
Name: Invalid username

InitialState			InputData			Output	
Restaurants		Log In	Restaurants		Log In	Restaurants	Log In
	Create Account			Create Account		Invalid account information (usernan	ne must only contain lowercase letters, numbers, dots, and underscores).
	First Name  Last Name  Username  Password  Create Account			John Smith john smith Create Account			Create Account  First Name  Last Name  Username  Password  Create Account

# 1.4. Create account (Duplicate username)

#### ExpectedOutput:

Message: Username already exists

Page: /account/create

InitialState:

LoggedInUser: null
Page: /account/create

InputData:

first\_name: Patrick
last\_name: Smith
password: password
username: patrick
Name: Duplicate username

IESI PASSED								
InitialState			InputData			Output		
Restaurants		Log In	Restaurants		Log In	Restaurants		Log In
	Create Account			Create Account		Username already exists.		
	Eirst Name  Last Name  Username  Password  Create Account			Patrick Smith patrick Create Account			First Name  Last Name  Username  Password  Create Account	

# 1.5. Create account (Valid)

ExpectedOutput:

Message: Account created

Page: /account
InitialState:

LoggedInUser: null
Page: /account/create

InputData:

first\_name: John
last\_name: Smith
password: password
username: john

Name: Valid

Restaurants Log In Restaurants Log In Restaurants Log In Restaurants My Orders My Account Create Account Account Created.	Log Out (John)
Create Account Create Account Created.	
First Name John	Account
Last Name Smith	eneral
Username	st Name
	hn
	st Name
	nith
	ssword
	ew password
В	lling Information
A	dress
	rd Number
	rd Expiry
	10 LAPII)
	rd Code
	Save

# 2. Modify account

Objective: Allow user to change account details.

Arrange: Open the modify account page (it will contain the existing information other than the password and card information).

Act: Fill in the new password, credit card (optional), and/or mailing address (optional).

Assert: If the information is valid (the password and card information can be empty, or be a new password), it should be updated and the user should be redirected back to the homepage.

#### 2.1. Modify account (Valid password change)

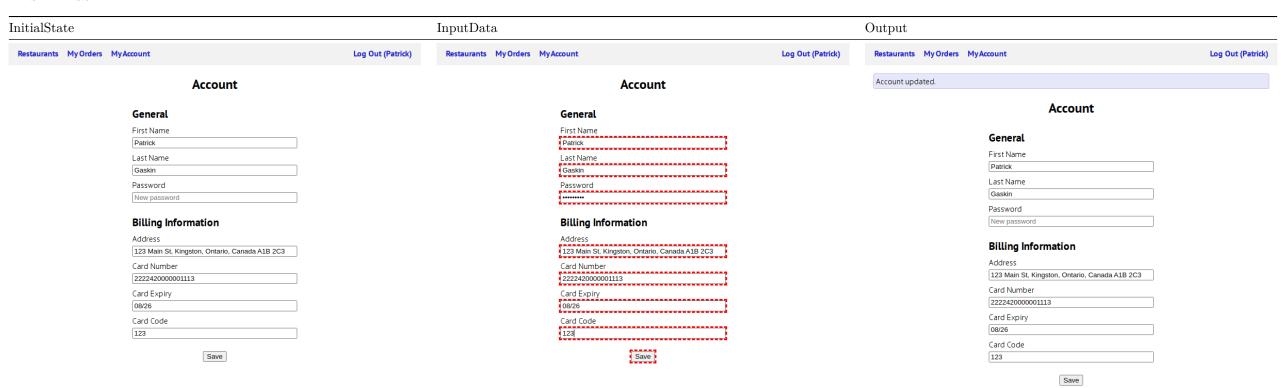
ExpectedOutput:
Message: Account updated
Page: /account
InitialState:
LoggedInUser: patrick
Page: /account
InputData:

address: 123 Main St, Kingston, Ontario, Canada A1B 2C3

cardcode: '123'
cardexpiry: 08/26

cardnumber: '2222420000001113'

firstname: Patrick
lastname: Gaskin
password: password1
Name: Valid password change



#### 2.2. Modify account (Invalid card information change)

ExpectedOutput:

Message: Invalid card information
Page: /account
InitialState:
LoggedInUser: patrick
Page: /account
InputData:
address: 123 Main St, Kingston, Ontario, Canada A1B 2C3
cardcode: 123
cardexpiry: 99/99
cardnumber: 1
firstname: Patrick
lastname: Gaskin
password: ''
Name: Invalid card information change

#### TEST PASSED

InitialState  ${\bf Input Data}$ Output Restaurants My Orders My Account Restaurants My Orders My Account Log Out (Patrick) Restaurants My Orders My Account Log Out (Patrick) Log Out (Patrick) Invalid card information (invalid credit card number). **Account** Account Account General General First Name First Name General Patrick Patrick First Name Last Name Last Name Patrick Gaskin Gaskin Last Name Gaskin New password Password **Billing Information Billing Information** Address 123 Main St, Kingston, Ontario, Canada A1B 2C3 **Billing Information** 123 Main St, Kingston, Ontario, Canada A1B 2C3 Address Card Number Card Number 123 Main St, Kingston, Ontario, Canada A1B 2C3 2222420000001113 Card Number Card Expiry Card Expiry 99/99 2222420000001113 08/26 Card Code Card Expiry Card Code 08/26 123 123 Card Code Save Save 123 Save

# 2.3. Modify account (Valid card information and password change)

ExpectedOutput:

Message: Account updated

Page: /account
InitialState:

LoggedInUser: patrick

Page: /account

InputData:

address: 123 Main St, Kingston, Ontario, Canada A1B 2C3

cardcode: 123
cardexpiry: 09/26

cardnumber: 2223000048410010

firstname: Patrick
lastname: Gaskin
password: password1

Name: Valid card information and password change

#### TEST PASSED

InitialState		InputData		Output	
Restaurants My Orders My Account	Log Out (Patrick)	Restaurants My Orders My Account	Log Out (Patrick)	Restaurants My Orders My Account	Log Out (Patrick)
Account		Account		Account updated.	
General		General		Account	
First Name Patrick  Last Name Gaskin  Password  New password  Billing Information  Address  123 Main St, Kingston, Ontario, Canada A1B 2C3  Card Number		First Name Patrick Last Name Gaskin Password  Billing Information Address 123 Main St, Kingston, Ontario, Canada A1B 2C3 Card Number		General First Name Patrick Last Name Gaskin Password New password  Billing Information Address 123 Main St, Kingston, Ontario, Canada A1B 2C3	
222242000001113  Card Expiry  08/26  Card Code  123  Save		2223000048410010   Card Expiry   99/26   Card Code   123    Save		Card Number  2223000048410010  Card Expiry  09/26  Card Code  123	

1

# 3. Log in

**Objective**: Authenticate a user to provide them access to the application.

**Arrange**: Open the account login page.

Act: Fill in login information (e.g., username, password) and click log in.

**Assert**: If the login information is valid, the user should be logged in and redirected to the homepage.

# 3.1. Log in (Incorrect username)

ExpectedOutput:

Message: User does not exist

Page: /account/login
InitialState:

LoggedInUser: null Page: /account/login

InputData:

password: password username: user

Name: Incorrect username

InitialState		InputData			Output		
Restaurants	Log In	Restaurants		Log In	Restaurants		Log In
L	og In		Log In		User does not exist.		
Username Password  Crea	Username user  Password user  Log In		user Log in Create Account			Log In  Username  Password  Log In  Create Account	

# 3.2. Log in (Incorrect password)

ExpectedOutput:

Message: Incorrect password

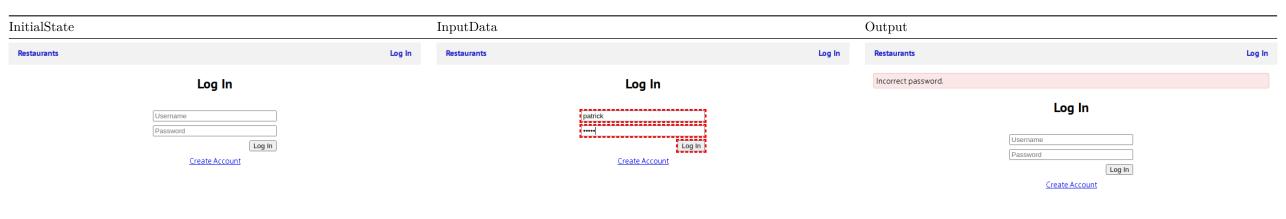
Page: /account/login

InitialState:

LoggedInUser: null
Page: /account/login

InputData:

password: dummy
username: patrick
Name: Incorrect password



# 3.3. Log in (Success)

ExpectedOutput:

LoggedInUser: patrick Page: /restaurants

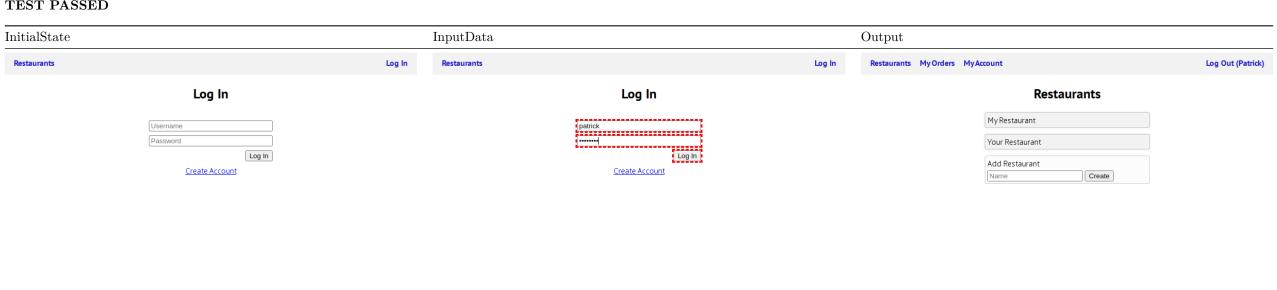
InitialState:

LoggedInUser: null Page: /account/login

InputData:

password: password username: patrick

Name: Success



# 4. Log out

**Objective**: Allow user to log out from the application.

Arrange: Open any page and click log out.

Act: N/A.

**Assert**: The user should be logged out and redirected to the login page.

# 4.1. Log out (Success)

ExpectedOutput:
LoggedInUser: null Page: /restaurants
InitialState:

LoggedInUser: patrick
Page: /account/logout

Name: Success

Restaurants  Restaurants  Restaurants  My Restaurant  My Restaurant  My Restaurant	ما urants
	urants
My Restaurant	
Try Nestablish	
Your Restaurant	

# 5. List customer orders

Objective: Allow user to view orders.

**Arrange**: Open the homepage and click the orders button.

 $\mathbf{Act}$ : N/A.

Assert: A summary (date, restaurant, total, status) of orders for the customer should be shown.

# 5.1. List customer orders (List customer orders)

ExpectedOutput:

VisibleOrderIDs:

**-** 2

InitialState:

LoggedInUser: patrick
Page: /orders

Name: List customer orders

# TEST PASSED

InitialState	InputData	Output
Restaurants My Orders My Account	Log Out (Patrick)	Restaurants My Orders My Account Log Out (Patrick)
My Orders		My Orders
My Restaurant #2 PAID 2023-10-04 20:19:32 123 Main St, Kingston, Ontario, Canada A1B 2C3 \$3.15		My Restaurant #2 PAID 2023-10-04 20:19:32 123 Main St, Kingston, Ontario, Canada A1B 2C3 \$3.15

N/A

# 6. List restaurants

**Objective**: Show all non-deleted restaurants on the app.

**Arrange**: Open the homepage.

Act: Click the restaurants button.

Assert: All non-deleted restaurants should be listed. There should also be a new restaurant button.

# 6.1. List restaurants (Display list of restaurants)

# ExpectedOutput: VisibleRestaurantIDs: - 1 - 2 InitialState: LoggedInUser: patrick

Page: /restaurants

Name: Display list of restaurants

TEST PASSED							
InitialState			InputData		Output		
Restaurants My Orders My Account Log Out (Patrick)			Restaurants My Orders N		count	Log Out (Patrick)	
	Restaurants					Restaurants	
	My Restaurant					My Restaurant	
	Your Restaurant					Your Restaurant	
	Add Restaurant Name Create					Add Restaurant  Name  Create	

#### 7. View restaurant

**Objective**: Show restaurant information.

Arrange: Open the restaurants page and click on a restaurant.

 $\mathbf{Act}$ : N/A.

Assert: Restaurant information and a preview of the menu items should be displayed. If the user is the owner, there should also be an employee list (with a remove button for each employee and a username input and add employee button), and the restaurant plus all menu items should have editable fields with a save/remove button, plus fields and a button to add a new menu item. If the user is an employee, there should be a section with all paid and accepted orders listed (not cancelled, pending, or delivered).

#### 7.1. View restaurant (Customer+Owner)

#### ExpectedOutput:

RestaurantName: My Restaurant

VisibleButtons:

- restaurant:order

- restaurant:delete

- restaurant:update

- employee:new:add

- employee:qays:remove

- item:new:add

- item:2:delete

- item:2:update

#### VisibleMenuItems:

- Tea

- Pop

- Water

#### InitialState:

LoggedInUser: patrick
Page: /restaurants/1
Name: Customer+Owner



# 7.2. View restaurant (Customer+Employee)

#### ExpectedOutput: RestaurantName: My Restaurant VisibleButtons: - restaurant:order - order:2:accept - order:1:deliver VisibleMenuItems: - Tea - Pop - Water VisibleOrderIDs: - 1 **-** 2 InitialState: LoggedInUser: qays Page: /restaurants/1 Name: Customer+Employee

alState		InputData	Output	
staurants My Orders My Account	Log Out (Qays)		Restaurants My Orders My Account	
My Restauran	t		My Restaurant	
Create Order			Create Order	
Menu			Menu	
Tea	\$1.20		Tea	
Pop	\$1.10		Рор	
Water	\$1.00		Water	
mployee: Orders			Employee: Orders	
Order #1 ACCEPTED 2023-10-04 20:19:32 123 Main St, Kingston, Ontario, Canada A1B 2C3 Deliver Details			Order #1 ACCEPTED 2023-10-04 20:19:32 123 Main St, Kingston, Ontario, Canada A1B 2C3 Deliver Details	
Order #2 PAID 2023-10-04 20:19:32 123 Main St, Kingston, Ontario, Canada A1B 2C3 Accept Details			Order #2 PAID 2023-10-04 20:19:32 123 Main St, Kingston, Ontario, Canada A1B 2C3 Accept Details	

# 7.3. View restaurant (Customer)

ExpectedOutput:

RestaurantName: My Restaurant

VisibleButtons:
- restaurant:order

VisibleMenuItems:

- Tea

- Pop - Water InitialState:

LoggedInUser: jeff
Page: /restaurants/1

Name: Customer

ialState	InputData	Output	
Restaurants My Orders My Account	Log Out (Jeff)	Restaurants My Orders My Account	
My Restaurant		My Restaurant	
Create Order		Create Order	
<b>1</b> enu		Menu	
Tea	\$1.20	Tea	
Pop	\$1.10	Pop	
Water	\$1.00	Water	

# 8. New order

Objective: Allow user to create a new order.

**Arrange**: Go to a restaurant page.

Act: Click new order.

Assert: The order page for a new pending order should be displayed with a list of available menu items and the current quantity (all zero).

# 8.1. New order (Valid new order)

ExpectedOutput:
Page: /orders/4
VisibleButtons:
- order:pay
- order:cancel
InitialState:
LoggedInUser: je:

LoggedInUser: jeff
Page: /restaurants/1

InputData:

action: restaurant:order
Name: Valid new order

InitialState			InputData		Output		
Restaurants My Orders My	Account	Log Out (Jeff)	Restaurants My Orders My Account	Log Out (Jeff)	Restaurants My Orders My Account	Log Out (Jeff)	
	My Restaurant		Му Б	estaurant	1	My Restaurant #4	
	Create Order		C	reate Order		Submit Order Cancel Order	
Menu			Menu		Tea	\$1.20 - 0 +	
Tea		\$1.20	Tea	\$1.20	Рор	\$1.10 - 0 +	
Pop		\$1.10	Pop	\$1.10	Water	\$1.00 - 0 +	
Water		\$1.00	Water	\$1.00			

# 9. Add/remove item in order

Objective: Allow user to add item to order.

**Arrange**: Open the order page for a pending order.

Act: Adds item to order.

**Assert**: The item quantity should be increased or decreased.

# 9.1. Add/remove item in order (Add)

ExpectedOutput:
OrderItems:
- Tea=1
- Pop=1
- Water=0
Page: /orders/3
InitialState:
LoggedInUser: qays
Page: /orders/3
InputData:

action: item:4:add

Name: Add

InitialState			InputData		Output		
Restaurants My Orders My Account Log Out (Qays)		Restaurants My Orders My Account Log Out (Qays)		Restaurants My Orders My Account	Log Out (Qays)		
	My Restaurant #3		My Restat	urant #3	1	My Restaurant #3	
	Submit Order Cancel Order		Submit Order	Cancel Order		Submit Order Cancel Order	
Tea		\$1.20 - 1 +	Tea	\$1.20 - 1 +	Tea	\$1.20 - 1 +	
Pop		\$1.10 - 0 +	Pop	\$1.10 - 0 +	Pop	\$1.10 - 1 +	
Water		\$1.00 - 0 +	Water	\$1.00 - 0 +	Water	\$1.00 - 0 +	

# 9.2. Add/remove item in order (Remove)

# ExpectedOutput: OrderItems: - Tea=0 - Pop=0 - Water=0 Page: /orders/3 InitialState: LoggedInUser: qays Page: /orders/3 InputData:

action: item:2:subtract

Name: Remove

InitialState		InputData		Output		
Restaurants My Orders My Account Log Out (Qays)		Log Out (Qays)	Restaurants My Orders My Account	Log Out (Qays)	Restaurants My Orders My Account	Log Out (Qays)
	My Restaurant #3		My Restaurant #3		My Restaurant #	3
	Submit Order Cancel Order		Submit Order Cancel Order		Submit Order Cancel Order	it .
Tea		\$1.20 - 1 +	Tea	\$1.20 1 +	Tea	\$1.20 - 0 +
Pop		\$1.10 - 0 +	Pop	\$1.10 - 0 +	Pop	\$1.10 - 0 +
Water		\$1.00 - 0 +	Water	\$1.00 - 0 +	Water	\$1.00 - 0 +

# 9.3. Add/remove item in order (Invalid)

# ExpectedOutput: OrderItems: - Tea=1 - Pop=0 - Water=0 Page: /orders/3 InitialState: LoggedInUser: qays Page: /orders/3 InputData:

force\_action: item:6:add

Name: Invalid



#### 10. Submit order

Objective: Allow user to submit an order.

**Arrange**: Open the order page for a pending order.

Act: Press the submit order button.

Assert: If the user has valid payment and mailing information, all selected items are not deleted, and the payment is successful, the order state should now be submitted, and the order page should show the order status and summary instead of the available menu items.

# 10.1. Submit order (Valid)

ExpectedOutput:

Message: Order submitted

Page: /orders/3
VisibleButtons: - order:cancel InitialState:

LoggedInUser: qays
Page: /orders/3
InputData:

action: order:pay

Name: Valid

InitialState		InputData		Output		
Restaurants My Orders My Account Log Out (Qays)		Restaurants My Orders My Account	Log Out (Qays)	Restaurants My Orders My Account	Log Out (Qays)	
	My Restaurant #3		My Restaurant #3		Order submitted.	
	Submit Order Cancel Order		Submit Order Cancel Order		My Restaurant #3	
Tea		\$1.20 - 1 +	Tea	\$1.20 - 1 +	Cancel Order Accept	
Pop		\$1.10 - 0 +	Pop	\$1.10 - 0 +	Tea	1
Water		\$1.00 - 0 +	Water	\$1.00 - 0 +		

# 11. Cancel order

**Objective**: Allow the user to cancel an order.

Arrange: Open the order page page for a paid or pending (but not accepted or delivered).

Act: Press the cancel order button.

Assert: If the order was in the paid or pending state, the order should now be in the cancelled state and the page should reflect that.

# 11.1. Cancel order (Invalid)

ExpectedOutput:

Message: Bad action OrderStatus: ACCEPTED Page: /orders/1 InitialState:

LoggedInUser: jeff Page: /orders/1
InputData:

force\_action: order:cancel

Name: Invalid

InitialState		InputData		Output	
Restaurants My Orders My Account	Log Out (Jeff)	Restaurants My Orders My Account	Log Out (Jeff)	Restaurants My Orders My Account	Log Out (Jeff)
My Restaurant #1		My Restaurant #1		Bad action.	
				My Restaurant #1	
Tea	1	Tea	1		
Slushie	5	Slushie	5	Tea	1
		FORCED ACTION: order.cancel		Slushie	5

# 11.2. Cancel order (Valid)

ExpectedOutput:

Message: Order cancelled OrderStatus: CANCELLED Page: /orders/2 InitialState:

LoggedInUser: patrick Page: /orders/2
InputData:

action: order:cancel

Name: Valid

InitialState			InputData		Output	
Restaurants My Orders	MyAccount	Log Out (Patrick)	Restaurants My Orders My Account	Log Out (Patrick)	Restaurants My Orders My Account	Log Out (Patrick)
	My Restaurant #2		My Restaurant #2		Order cancelled.	
	Cancel Order		Cancel Order		My Restaurant #2	
Pop		4	Pop	4		
Water		8	Water	8	Pop	4
					Water	8

## 11.3. Cancel order (Valid)

ExpectedOutput:

Message: Order cancelled OrderStatus: CANCELLED Page: /orders/3 InitialState:

LoggedInUser: qays
Page: /orders/3
InputData:

action: order:cancel

Name: Valid

InitialState			InputData		Output	
Restaurants My Orders My Accou	int	Log Out (Qays)	Restaurants My Orders My Account	Log Out (Qays)	Restaurants My Orders My Account	Log Out (Qays)
	My Restaurant #3		My Restaurant #3		Order cancelled.	
	Submit Order Cancel Order		Submit Order Cancel Order		My Restaurant #3	
Tea		\$1.20 - 1 +	Tea	\$1.20 - 1 +		
Pop		\$1.10 - 0 +	Pop	\$1.10 - 0 +	Tea	1
Water		\$1.00 - 0 +	Water	\$1.00 - 0 +		

## 12. Create restaurant

Objective: Allow owners to add restaurant on the app.

**Arrange**: Go to the restaurants.

Act: Fill in a restaurant name and press add.

Assert: The user should be redirected to the restaurants page, and the new restaurant should be listed.

## 12.1. Create restaurant (Invalid)

ExpectedOutput:

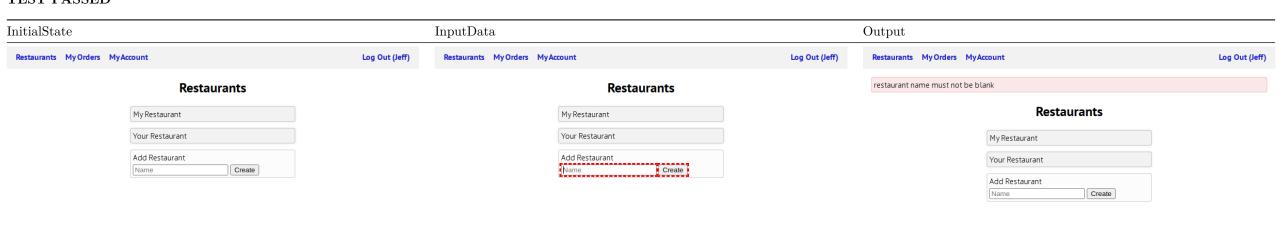
Message: must not be blank

Page: /restaurants

InitialState:

LoggedInUser: jeff Page: /restaurants

InputData:
 name: ''
Name: Invalid



## 12.2. Create restaurant (Valid)

## ExpectedOutput:

Message: Restaurant created

Page: /restaurants

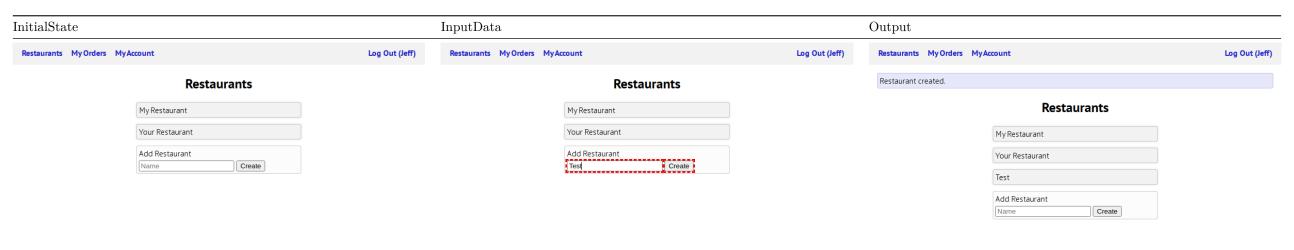
RestaurantListContainsName:

- Test

InitialState:

LoggedInUser: jeff
Page: /restaurants

InputData:
 name: Test
Name: Valid



## 13. Update restaurant

**Objective**: Allow owners to update the restaurant name.

**Arrange**: Open the page of a restaurant as an owner.

Act: Fill in a new name and submit the page.

Assert: The user should be redirected back to the restaurant page, and the name should be updated.

## 13.1. Update restaurant (Success)

ExpectedOutput:

Page: /restaurants/1

RestaurantName: Best Restaurant

InitialState:

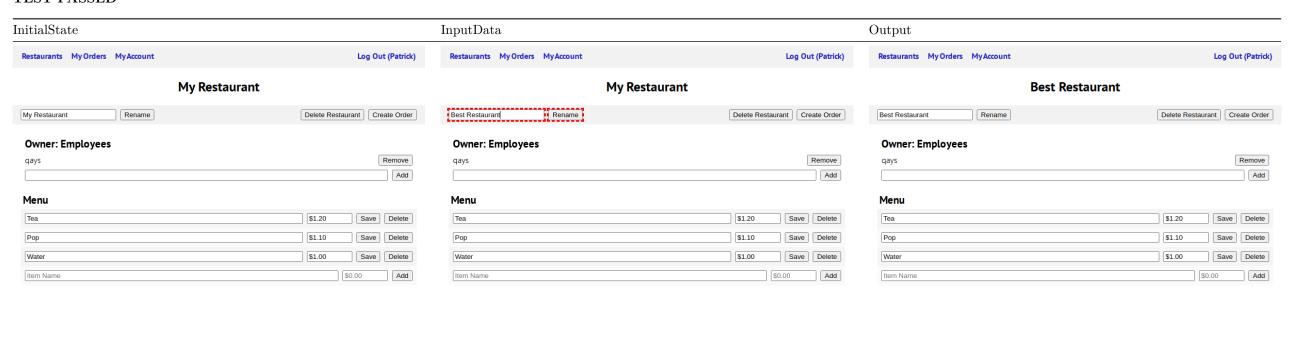
LoggedInUser: patrick
Page: /restaurants/1

InputData:

action: restaurant:update

restaurant:name: Best Restaurant

Name: Success



## 14. Delete restaurant

**Objective**: Allow owners to delete the restaurant.

**Arrange**: Open the page of a restaurant as an owner.

Act: Press the delete button.

Assert: The user should be redirected back to the restaurants list, and the restaurant should no longer be present.

## 14.1. Delete restaurant (Success)

## ExpectedOutput:

NotVisibleRestaurantNames:

- My Restaurant

Page: /restaurants

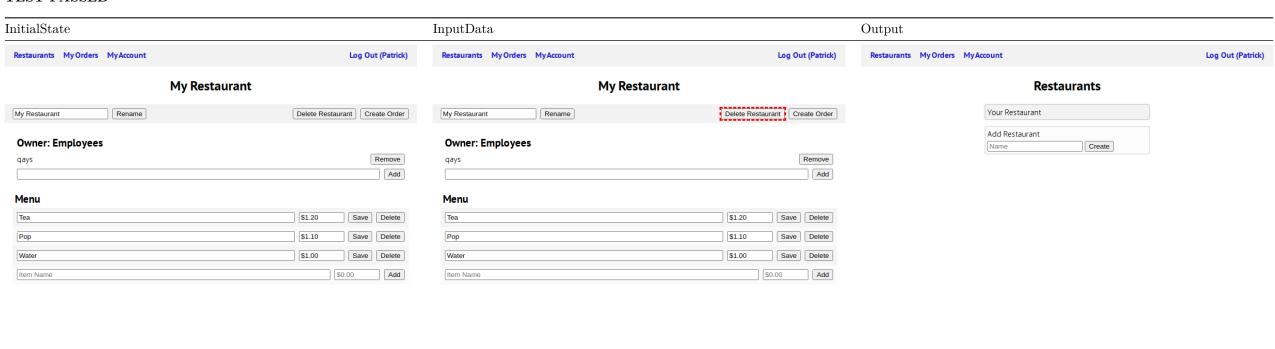
InitialState:

LoggedInUser: patrick
Page: /restaurants/1

InputData:

action: restaurant:delete

Name: Success



## 15. Add employee

Objective: Add new employee to restaurant.

**Arrange**: Open the page of a restaurant as an owner.

Act: Fill in a valid username for a new employee and press the add employee button.

Assert: If the username exists, the user should be redirected back to the restaurant page and the new employee should be added.

## 15.1. Add employee (Invalid)

## ExpectedOutput:

Message: user does not exist
Page: /restaurants/1
VisibleEmployeeUsernames:
- qays

InitialState:

LoggedInUser: patrick
Page: /restaurants/1

InputData:

action: employee:new:add
employee:new:username: sdfsdf

Name: Invalid



## 15.2. Add employee (Valid)

## ExpectedOutput: Page: /restaurants/1 VisibleEmployeeUsernames: - qays - jeff InitialState: LoggedInUser: patrick Page: /restaurants/1 InputData: action: employee:new:add employee:new:username: jeff Name: Valid



## 16. Remove employee

Objective: Remove employee from restaurant.

**Arrange**: Open the page of a restaurant as an owner.

Act: Press the remove employee button beside an employee.

**Assert**: Check that employee was removed.

## 16.1. Remove employee (Success)

ExpectedOutput:

Page: /restaurants/1

VisibleEmployeeUsernames: []

InitialState:

LoggedInUser: patrick
Page: /restaurants/1

InputData:

action: employee:qays:remove

Name: Success



## 17. Add menu item

Objective: Allow owners to add menu items.

**Arrange**: Open the page of a restaurant as an owner.

Act: Fill in the name and price for a new menu item, and press the add item button.

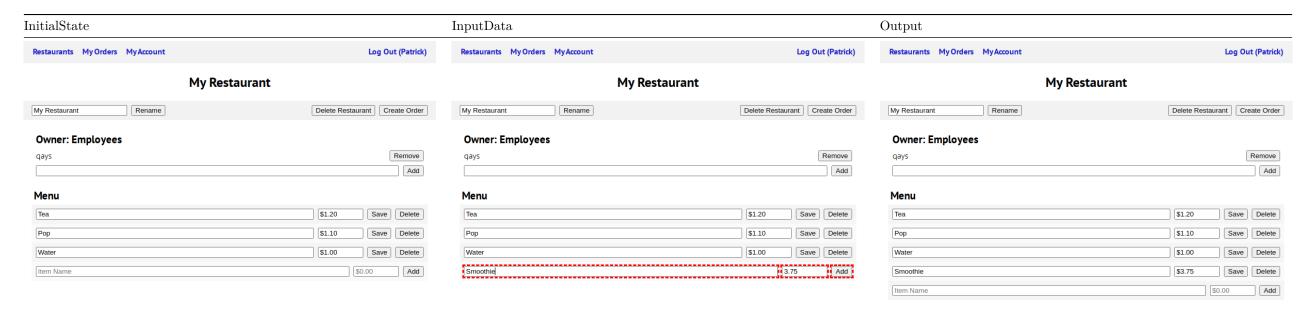
Assert: If the information provided is valid, the user should be redirected back to the restaurant page and the item should be added to the menu.

## 17.1. Add menu item (Valid)

## ExpectedOutput: Page: /restaurants/1 VisibleMenuItems: - Tea - Pop - Water - Smoothie InitialState: LoggedInUser: patrick Page: /restaurants/1 InputData: action: item:new:add item:new:name: Smoothie item:new:price: 3.75

## TEST PASSED

Name: Valid



## 17.2. Add menu item (Invalid)

ExpectedOutput:

Message: invalid price
Page: /restaurants/1
VisibleMenuItems:
- Tea
- Pop
- Water
InitialState:
LoggedInUser: patrick
Page: /restaurants/1
InputData:
action: item:new:add
item:new:name: dummy
item:new:price: -3.75
Name: Invalid



## 17.3. Add menu item (Invalid)

ExpectedOutput:
 Message: invalid price
 Page: /restaurants/1
 VisibleMenuItems:
 - Tea
 - Pop
 - Water
InitialState:
 LoggedInUser: patrick
 Page: /restaurants/1
InputData:
 action: item:new:add
 item:new:name: dummy
 item:new:price: one dollar
Name: Invalid

CHECK FAILURE: Message('invalid price') - incorrect message 'could not convert string to float: 'one dollar"

## TEST FAILED



## 17.4. Add menu item (Invalid)

# ExpectedOutput: Message: item name Page: /restaurants/1 VisibleMenuItems: - Tea - Pop - Water InitialState: LoggedInUser: patrick Page: /restaurants/1 InputData: action: item:new:add item:new:name: '' item:new:price: 1.0 Name: Invalid



## 18. Update menu item

Objective: Allow owner to update menu item names and prices.

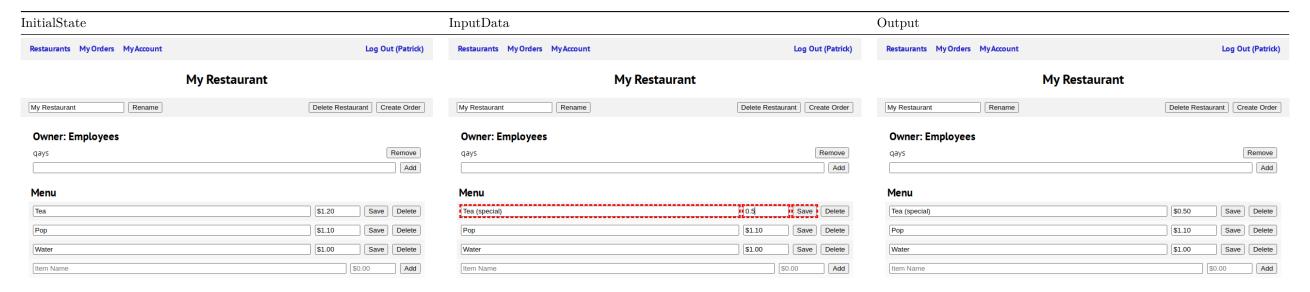
**Arrange**: Open the page of the restaurant as an owner.

Act: Update the information for a menu item, and press the save button.

Assert: If the information provided is valid, the user should be redirected back to the restaurant page and the item should be updated.

## 18.1. Update menu item (Valid)

## ExpectedOutput: Page: /restaurants/1 VisibleMenuItems: - Tea (special) - Pop - Water InitialState: LoggedInUser: patrick Page: /restaurants/1 InputData: action: item:2:update item:2:name: Tea (special) item:2:price: 0.5 Name: Valid



## 18.2. Update menu item (Invalid)

```
ExpectedOutput:
    Message: item name must not be empty
    Page: /restaurants/1
    VisibleMenuItems:
        - Tea
        - Pop
        - Water
InitialState:
    LoggedInUser: patrick
    Page: /restaurants/1
InputData:
    action: item:2:update
    item:2:name: ''
    item:2:price: 1.2
Name: Invalid
```



## 19. Delete menu item

Objective: Allow owner to delete menu items.

**Arrange**: Open the page of the restaurant as an owner.

Act: Press the delete button beside a menu item.

**Assert**: Check that menu information is deleted.

## 19.1. Delete menu item (Valid)

ExpectedOutput:

Page: /restaurants/1 VisibleMenuItems: - Pop - Water InitialState:

LoggedInUser: patrick Page: /restaurants/1

InputData:

action: item:2:delete

Name: Valid



## 20. Accept restaurant order

**Objective**: Allow employees to update customer orders.

**Arrange**: Open the page of the restaurant as an owner.

Act: Press the accept button beside a paid order.

Assert: The order status should change to accepted, and the deliver button should appear.

## 20.1. Accept restaurant order (Success)

ExpectedOutput:

Message: Order accepted
Page: /restaurants/1
VisibleButtons:
- order:2:deliver

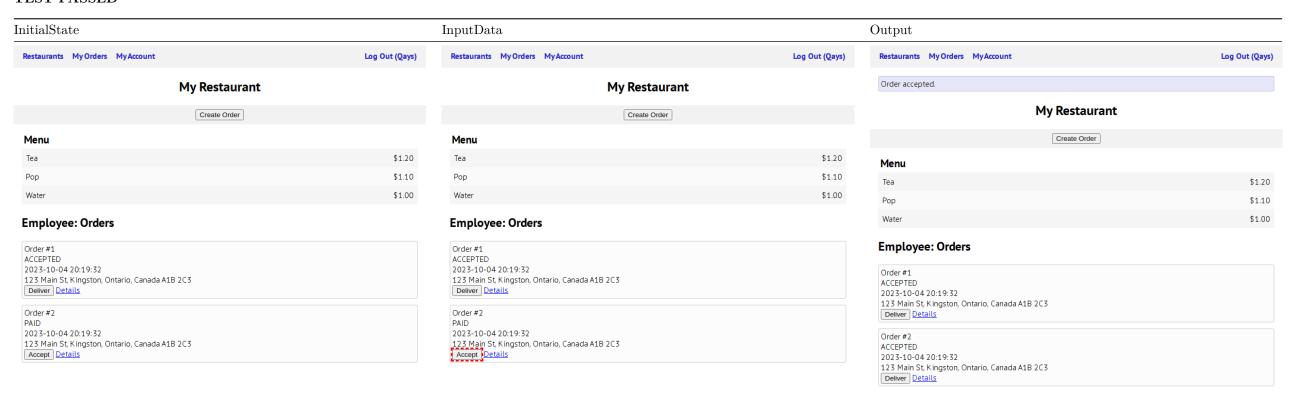
InitialState:

LoggedInUser: qays
Page: /restaurants/1

InputData:

action: order:2:accept

Name: Success



## 21. Deliver restaurant order

**Objective**: Allow employees to update customer orders.

**Arrange**: Open the page of the restaurant as an owner.

Act: Press the deliver button beside an accepted order.

**Assert**: The order should be removed from the list.

## 21.1. Deliver restaurant order (Success)

ExpectedOutput:

Message: Order delivered NotVisibleOrderIDs:

- 1

Page: /restaurants/1

InitialState:

LoggedInUser: qays
Page: /restaurants/1

InputData:

action: order:1:deliver

Name: Success

