

# Bristol Regional Food Network

## Digital Marketplace Platform – Test Cases

### Introduction

This document provides comprehensive test cases for the Bristol Regional Food Network digital marketplace platform. These test cases are designed to validate system functionality from stakeholder perspectives, focusing on user stories and acceptance criteria rather than technical implementation details.

The test cases cover all key stakeholder groups: Producers, Customers (including young professionals, families, community groups, and restaurants), and System requirements.

### Test Case Priority Guide

Test cases are designed based on user stories and categorised by priority to help plan sprint development:

**Critical Priority:** Core functionality required for minimum viable product. Should be implemented in Sprint 1 and early Sprint 2. System cannot function without these features.

**High Priority:** Important features that significantly enhance user experience and system utility. Should be implemented in Sprint 2 and early Sprint 3.

**Medium Priority:** Valuable features that add functionality and meet specific stakeholder needs. Should be implemented in Sprint 3 if time permits.

**Low Priority:** Enhancement features that improve community engagement but are not essential for core marketplace operations.

### Testing Approach

Testing should be approached as follows:

1. Test cases focus on user-facing functionality and business logic, not just technical implementation choices. Features may be implemented using different technical approaches (primarily using Django and Docker) as long as acceptance criteria are met.
2. Payment processing should use test environments (Stripe Test Mode, PayPal Sandbox, or mock services). **Never use real payment systems or actual financial data.**
3. Test cases include edge cases and error handling scenarios. The developed system should handle both success and failure paths gracefully.
4. Security test cases (particularly **TC-022**) are mandatory and must be demonstrated regardless of priority. Authentication and authorisation are non-negotiable requirements.

## Test Cases

**TC-001: As a producer, I want to create an account so that I can list my products on the marketplace.**

<b>Test Case ID</b>	TC-001
<b>User Story</b>	As a producer, I want to create an account so that I can list my products on the marketplace.
<b>Stakeholder</b>	Producer
<b>Priority</b>	Critical
<b>Description</b>	Validates that producers can successfully register for the marketplace platform with required business information and receive appropriate authentication credentials.
<b>Preconditions</b>	The system is accessible and running No existing account with the test email address
<b>Test Steps</b>	<ol style="list-style-type: none"><li>1. Navigate to the producer registration page</li><li>2. Enter business name: Bristol Valley Farm</li><li>3. Enter contact name: Jane Smith</li><li>4. Enter email: jane.smith@bristolvalleyfarm.com</li><li>5. Enter phone: 01179 123456</li><li>6. Enter business address and postcode: BS1 4DJ</li><li>7. Enter password meeting security requirements</li><li>8. Confirm password</li><li>9. Submit registration form</li></ol>
<b>Expected Results</b>	Registration form accepts all valid inputs without error Account is created successfully with producer role Producer receives confirmation message Producer can log in with registered credentials Producer profile is accessible with entered business information
<b>Acceptance Criteria</b>	Producer account is created in the system Password is securely stored (hashed) Business information is correctly saved Producer can authenticate using email and password Appropriate producer permissions are assigned to the account

**TC-002: As a customer, I want to register for an account so that I can browse and purchase local products.**

<b>Test Case ID</b>	TC-002
<b>User Story</b>	As a customer, I want to register for an account so that I can browse and purchase local products.
<b>Stakeholder</b>	Customer (Young Professional/Family)

<b>Priority</b>	Critical
<b>Description</b>	Validates that customers can successfully create accounts with personal information and delivery address details for purchasing purposes.
<b>Preconditions</b>	The system is accessible and running No existing account with the test email address
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the customer registration page</li> <li>2. Enter full name: Robert Johnson</li> <li>3. Enter email: robert.johnson@email.com</li> <li>4. Enter phone: 07700 900123</li> <li>5. Enter delivery address: 45 Park Street, Bristol</li> <li>6. Enter postcode: BS1 5JG</li> <li>7. Enter password meeting security requirements</li> <li>8. Confirm password</li> <li>9. Accept terms and conditions</li> <li>10. Submit registration form</li> </ol>
<b>Expected Results</b>	Registration form accepts all valid inputs Customer account is created successfully Customer receives confirmation message Customer can log in with registered credentials Delivery address is stored for future orders
<b>Acceptance Criteria</b>	Customer account is created with appropriate role Personal information is securely stored Delivery address is saved and linked to customer profile Customer can authenticate successfully Customer has browsing and purchasing permissions

**TC-003: As a producer, I want to list a new product so that customers can find and purchase it.**

<b>Test Case ID</b>	TC-003
<b>User Story</b>	As a producer, I want to list a new product so that customers can find and purchase it.
<b>Stakeholder</b>	Producer
<b>Priority</b>	Critical
<b>Description</b>	Validates that authenticated producers can successfully create product listings with all required information including seasonal availability.
<b>Preconditions</b>	Producer is logged in to the system Producer has a verified account
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the product management section</li> </ol>

	<ol style="list-style-type: none"> <li>Click 'Add New Product'</li> <li>Enter product name: Organic Free Range Eggs</li> <li>Select category: Dairy &amp; Eggs</li> <li>Enter detailed description: Fresh organic eggs from free-range hens, collected daily</li> <li>Enter price: £3.50 per dozen</li> <li>Enter unit: Dozen</li> <li>Set availability: In Season (Available)</li> <li>Enter stock quantity: 50</li> <li>Add allergen information: Contains eggs</li> <li>Set harvest date: Current date</li> <li>Upload product image (optional)</li> <li>Submit product listing</li> </ol>
<b>Expected Results</b>	<p>Product form accepts all valid inputs</p> <p>Product is created and saved to the database</p> <p>Product appears in producer's product management dashboard</p> <p>Product becomes visible to customers in the marketplace</p> <p>All product details are correctly displayed</p>
<b>Acceptance Criteria</b>	<p>Product is linked to the authenticated producer</p> <p>All required fields are validated and stored</p> <p>Product appears in search and category browsing</p> <p>Seasonal availability status is correctly displayed</p> <p>Stock quantity is tracked for inventory management</p>

**TC-004: As a customer, I want to browse products by category so that I can find specific items I need.**

<b>Test Case ID</b>	TC-004
<b>User Story</b>	As a customer, I want to browse products by category so that I can find specific items I need.
<b>Stakeholder</b>	Customer (All types)
<b>Priority</b>	Critical
<b>Description</b>	Validates that customers can effectively browse products organised by categories such as vegetables, dairy, bakery, preserves, and seasonal specialties.
<b>Preconditions</b>	<p>Multiple products exist in different categories</p> <p>At least 5 products in the Vegetables category</p> <p>At least 3 products in the Dairy category</p>
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>Navigate to the marketplace homepage</li> <li>View the category navigation menu</li> <li>Click on 'Vegetables' category</li> <li>Observe displayed products</li> <li>Return to main categories</li> </ol>

<b>Expected Results</b>	6. Click on 'Dairy Products' category 7. Observe displayed products 8. Verify each product shows key information: name, price, farm origin, availability
	Category navigation is clearly visible Clicking 'Vegetables' displays only vegetable products Clicking 'Dairy Products' displays only dairy products Each category shows appropriate products with correct categorization Products display name, price, producer name, and availability status Category pages load without errors
<b>Acceptance Criteria</b>	Products are correctly categorised Category filtering works accurately Only products marked as 'Available' or 'In Season' are displayed Product information is complete and readable Navigation between categories is intuitive

**TC-005: As a customer, I want to search for specific products so that I can quickly find what I need.**

<b>Test Case ID</b>	TC-005
<b>User Story</b>	As a customer, I want to search for specific products so that I can quickly find what I need.
<b>Stakeholder</b>	Customer (All types)
<b>Priority</b>	High
<b>Description</b>	Validates that the search functionality enables customers to locate products efficiently using product names, descriptions, or producer names.
<b>Preconditions</b>	Multiple products exist in the database Products have descriptive names and detailed descriptions Search functionality is enabled
<b>Test Steps</b>	1. Navigate to the marketplace homepage 2. Locate the search bar 3. Enter search term: tomatoes 4. Submit search 5. Observe search results 6. Clear search 7. Enter search term: organic 8. Submit search 9. Observe search results including products from different categories 10. Test edge case: Enter non-existent product name 11. Observe empty results handling
<b>Expected Results</b>	Search bar is visible and accessible

	<p>Searching for 'tomatoes' returns all tomato products</p> <p>Searching for 'organic' returns all products with 'organic' in name or description</p> <p>Search results display product name, price, producer, and category</p> <p>Search for non-existent items shows appropriate 'no results found' message</p> <p>Search is case-insensitive</p>
<b>Acceptance Criteria</b>	<p>Search queries return relevant results based on product name</p> <p>Search queries return relevant results based on product description</p> <p>Search functionality handles partial matches appropriately</p> <p>Empty search results are handled gracefully</p> <p>Search performance is acceptable</p>

**TC-006: As a customer, I want to add products to my shopping cart so that I can purchase multiple items together.**

<b>Test Case ID</b>	TC-006
<b>User Story</b>	As a customer, I want to add products to my shopping cart so that I can purchase multiple items together.
<b>Stakeholder</b>	Customer (All types)
<b>Priority</b>	Critical
<b>Description</b>	Validates that customers can add products to a shopping cart, modify quantities, and view cart contents before proceeding to checkout.
<b>Preconditions</b>	<p>Customer is logged in</p> <p>Multiple products are available for purchase</p> <p>Shopping cart functionality is enabled</p>
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Browse or search for a product: Organic Carrots</li> <li>2. Click on the product to view details</li> <li>3. Select quantity: 2 kg</li> <li>4. Click 'Add to Cart'</li> <li>5. Observe confirmation message</li> <li>6. Navigate to another product: Fresh Milk</li> <li>7. Select quantity: 3 litres</li> <li>8. Click 'Add to Cart'</li> <li>9. Click on cart icon to view cart contents</li> <li>10. Verify both products appear with correct quantities and prices</li> <li>11. Modify quantity of Organic Carrots to 3 kg</li> <li>12. Observe updated total price</li> </ol>
<b>Expected Results</b>	<p>Products can be added to cart successfully</p> <p>Cart icon displays item count</p> <p>Cart page shows all added products with correct details</p> <p>Individual item prices and total cart price are calculated correctly</p> <p>Quantity modifications update prices in real-time</p>

	Cart persists during the browsing session
<b>Acceptance Criteria</b>	Cart maintains state for logged-in customers Item quantities can be modified Price calculations are accurate Cart displays producer information for multi-vendor awareness Items can be removed from cart

**TC-007: As a customer, I want to place an order from a single producer so that I can purchase their products.**

<b>Test Case ID</b>	TC-007
<b>User Story</b>	As a customer, I want to place an order from a single producer so that I can purchase their products.
<b>Stakeholder</b>	Customer (All types)
<b>Priority</b>	Critical
<b>Description</b>	Validates the complete checkout process for orders containing products from a single producer, including payment processing and order confirmation.
<b>Preconditions</b>	Customer is logged in with delivery address Cart contains products from only one producer Payment system is configured (test mode) Producer has at least 48 hours lead time setting
<b>Test Steps</b>	1. Add products from 'Bristol Valley Farm' to cart 2. Navigate to cart page 3. Review cart contents confirming single producer 4. Click 'Proceed to Checkout' 5. Verify delivery address is pre-filled 6. Select delivery date (minimum 48 hours from now) 7. Review order summary showing producer details 8. Choose payment method 9. Enter test payment details (use test sandbox credentials) 10. Confirm order
<b>Expected Results</b>	Checkout page loads with correct cart contents Delivery address is displayed and can be edited Delivery date selector enforces minimum 48-hour lead time Order summary shows itemised products, subtotal, and 5% commission Payment is processed successfully (test mode) Order confirmation page displays with order number Customer receives order confirmation details Producer receives notification of new order
<b>Acceptance Criteria</b>	Order is created in the database with 'Pending' status Payment transaction is recorded

	5% network commission is calculated and recorded Producer payment (95% of order value) is calculated Order details include delivery date and customer information Both customer and producer can view the order
--	--

**TC-008: As a customer, I want to place an order from multiple producers so that I can purchase various products in one transaction.**

<b>Test Case ID</b>	TC-008
<b>User Story</b>	As a customer, I want to place an order from multiple producers so that I can purchase various products in one transaction.
<b>Stakeholder</b>	Customer (All types)
<b>Priority</b>	Critical
<b>Description</b>	Validates the multi-vendor checkout process including clear separation of producer responsibilities, individual delivery arrangements, and payment distribution.
<b>Preconditions</b>	Customer is logged in Cart contains products from at least 2 different producers Payment system is configured (test mode)
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Add 2 products from 'Bristol Valley Farm' to cart</li> <li>2. Add 2 products from 'Hillside Dairy' to cart</li> <li>3. Navigate to cart</li> <li>4. Verify products are grouped by producer</li> <li>5. Click 'Proceed to Checkout'</li> <li>6. Review order summary showing separate sections for each producer</li> <li>7. Note individual producer delivery information</li> <li>8. Select delivery dates for each producer (may differ based on their lead time)</li> <li>9. Review total cost breakdown showing per-producer subtotals</li> <li>10. Enter payment details</li> <li>11. Confirm multi-vendor order</li> </ol>
<b>Expected Results</b>	Cart clearly groups products by producer Checkout displays separate sections for each producer Each producer section shows their delivery requirements and timeline Order can have different delivery dates per producer Payment is split appropriately with 5% commission calculated on total Each producer (95% of their items) receives separate payment allocation Order confirmation shows clear breakdown by producer Each producer receives notification only for their portion
<b>Acceptance Criteria</b>	Multi-vendor order is recorded as single customer order Individual producer sub-orders are created and linked



	Payment distribution is calculated correctly per producer Network commission (5%) is calculated on total order value Each producer can view only their relevant order items Customer can track all items in one order view
--	---

**TC-009: As a producer, I want to view incoming orders so that I can prepare products for delivery.**

<b>Test Case ID</b>	TC-009
<b>User Story</b>	As a producer, I want to view incoming orders so that I can prepare products for delivery.
<b>Stakeholder</b>	Producer
<b>Priority</b>	Critical
<b>Description</b>	Validates that producers can access a dashboard showing all incoming orders with adequate lead time and complete customer details for preparation and delivery.
<b>Preconditions</b>	Producer is logged in At least 3 orders have been placed for this producer's products Orders have different delivery dates
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Log in as producer</li> <li>2. Navigate to 'Orders' or 'Order Management' section</li> <li>3. View list of incoming orders</li> <li>4. Observe order display includes: order number, customer name, order date, delivery date, items ordered, total value</li> <li>5. Click on a specific order to view full details</li> <li>6. Review detailed order showing: customer contact information, delivery address, itemised product list with quantities, special instructions (if any)</li> <li>7. Check that orders are sorted by delivery date</li> <li>8. Verify all orders show minimum 48-hour lead time from order date to delivery date</li> </ol>
<b>Expected Results</b>	Producer can access order management dashboard All orders for producer's products are visible Orders display essential preparation information Delivery dates respect 48-hour minimum lead time Producer can view complete customer details for each order Orders can be filtered or sorted by delivery date, status Order statuses are clearly indicated (Pending, Confirmed, Ready, Delivered)
<b>Acceptance Criteria</b>	Only orders containing producer's products are visible Multi-vendor orders show only this producer's items Customer contact information is accessible for delivery coordination Order information is accurate and complete

	Producer cannot access other producers' orders
--	--

**TC-010: As a producer, I want to update the status of orders so that customers know when their products are ready.**

<b>Test Case ID</b>	TC-010
<b>User Story</b>	As a producer, I want to update the status of orders so that customers know when their products are ready.
<b>Stakeholder</b>	Producer
<b>Priority</b>	High
<b>Description</b>	Validates that producers can update order statuses through the order lifecycle and customers receive appropriate notifications.
<b>Preconditions</b>	Producer is logged in At least one order exists with 'Pending' status Order has passed 48-hour preparation window
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to order management dashboard</li> <li>2. Select an order with 'Pending' status</li> <li>3. Click 'Update Status' or similar action</li> <li>4. Change status to 'Confirmed'</li> <li>5. Add optional note: Products will be prepared by delivery date</li> <li>6. Save status update</li> <li>7. Observe order now shows 'Confirmed' status</li> <li>8. Later, update same order status to 'Ready for Collection/Delivery'</li> <li>9. Save status update</li> <li>10. Verify customer receives notification (check customer account)</li> </ol>
<b>Expected Results</b>	Producer can change order status Status updates are saved and reflected immediately Order history shows status change timeline Optional notes can be added to status updates Customer sees updated status in their order view Appropriate notifications are triggered for status changes Status changes are logged with timestamp and producer information
<b>Acceptance Criteria</b>	Order status follows logical progression (Pending → Confirmed → Ready → Delivered) Status cannot skip required stages Customer view updates to reflect new status Audit trail maintains all status changes Only the relevant producer can update their order portion

**TC-011: As a producer, I want to update my product inventory so that customers only see available products I have.**

<b>Test Case ID</b>	TC-011
<b>User Story</b>	As a producer, I want to update my product inventory so that customers only see available products I have.
<b>Stakeholder</b>	Producer
<b>Priority</b>	High
<b>Description</b>	Validates that producers can modify product availability and stock quantities without requiring technical expertise.
<b>Preconditions</b>	Producer is logged in Producer has at least 3 listed products Products have various stock levels
<b>Test Steps</b>	1. Navigate to product management dashboard 2. View list of all producer's products 3. Select a product: Organic Tomatoes (current stock: 20 kg) 4. Click 'Edit' or 'Update Stock' 5. Change stock quantity to 35 kg 6. Change availability status to 'In Season' 7. Save changes 8. Select another product that is out of stock 9. Change availability status to 'Out of Season' or 'Unavailable' 10. Save changes 11. Verify updated products reflect changes in customer marketplace view
<b>Expected Results</b>	Producer can access and edit all their products Stock quantity updates save successfully Availability status updates save successfully Products marked 'Unavailable' are hidden from customer browsing Products marked 'In Season' appear in customer searches Stock quantity displays correctly to customers Changes take effect immediately
<b>Acceptance Criteria</b>	Producer can only edit their own products Stock updates are validated (non-negative numbers) Availability changes immediately affect customer visibility Product update history is maintained Updates trigger appropriate inventory alerts if stock is low

**TC-012: As a producer, I want to receive weekly payment settlements so that I can manage my business finances.**

<b>Test Case ID</b>	TC-012
<b>User Story</b>	As a producer, I want to receive weekly payment settlements so that I can manage my business finances.
<b>Stakeholder</b>	Producer

<b>Priority</b>	Critical
<b>Description</b>	Validates the payment settlement system that distributes 95% of order value to producers weekly with transparent audit trails.
<b>Preconditions</b>	Producer has completed orders from the previous week Payment week has ended (Sunday midnight) Orders have been delivered/completed Payment system is configured
<b>Test Steps</b>	1. Producer logs in on Monday after settlement week 2. Navigate to 'Payments' or 'Financial Reports' section 3. View weekly payment summary for completed week 4. Observe payment details showing: total orders value, 5% network commission deducted, 95% producer payment amount, individual order breakdown 5. Download payment report for tax records 6. Verify payment report includes: order numbers, customer names (anonymised if needed), product items sold, dates, amounts, commission breakdown 7. Check that payment status shows 'Processed' or 'Pending Bank Transfer' 8. Verify running total for the tax year is displayed
<b>Expected Results</b>	Weekly payment summary is accessible and accurate All completed orders from the week are included 5% commission is correctly calculated and deducted 95% producer payment is correctly calculated Payment report is downloadable in PDF or CSV format Report includes all necessary information for tax reporting Payment transaction ID or reference is provided Historical payment records are accessible
<b>Acceptance Criteria</b>	Payment calculations are accurate to 2 decimal places Only completed/delivered orders are included in settlements Payment reports are suitable for business accounting and tax filing Settlement timeline adheres to weekly schedule Producer can track payment status Audit trail links payments to specific orders

**TC-013: As a customer, I want to view food miles for products so that I can make environmentally conscious purchases.**

<b>Test Case ID</b>	TC-013
<b>User Story</b>	As a customer, I want to view food miles for products so that I can make environmentally conscious purchases.
<b>Stakeholder</b>	Customer (All types)
<b>Priority</b>	Medium

<b>Description</b>	Validates the environmental reporting feature that calculates and displays food miles based on customer postcode and producer farm location.
<b>Preconditions</b>	Customer is logged in with postcode on record Customer postcode: BS1 5JG Producer farm locations are recorded in the system Products are associated with producer locations
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Browse products in any category</li> <li>2. View a product from 'Bristol Valley Farm' (located at BS1 4DJ)</li> <li>3. Observe food miles display on product details</li> <li>4. Note the calculated distance in miles</li> <li>5. Browse a product from a producer further away (e.g., 15 miles)</li> <li>6. Compare food miles between products</li> <li>7. Add both products to cart</li> <li>8. View cart page</li> <li>9. Observe total food miles calculation for the order</li> <li>10. Verify calculation is sum of individual product distances</li> </ol>
<b>Expected Results</b>	Food miles are displayed for each product Distance calculation is based on straight-line or road distance between customer postcode and producer location Food miles are shown prominently (e.g., badge, icon with number) Products from closer producers show lower food miles Cart view displays cumulative food miles for all items Food miles contribute to sustainability awareness
<b>Acceptance Criteria</b>	Distance calculation is accurate based on postcodes All products display food miles information Food miles update if customer changes delivery address Visual representation is clear and easy to understand Calculation supports the network's 20-mile radius commitment

**TC-014: As a customer, I want to filter products by organic certification so that I can find certified organic items.**

<b>Test Case ID</b>	TC-014
<b>User Story</b>	As a customer, I want to filter products by organic certification so that I can find certified organic items.
<b>Stakeholder</b>	Customer (Young Professional/Family)
<b>Priority</b>	Medium
<b>Description</b>	Validates quality assurance features allowing customers to filter and identify products with organic certification status.
<b>Preconditions</b>	Multiple products exist with varying organic certification statuses At least 5 products are marked as 'Certified Organic' At least 5 products are marked as 'Not Certified'

<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to product browsing page</li> <li>2. Locate filtering options</li> <li>3. Select 'Organic Certification' filter</li> <li>4. Enable filter for 'Certified Organic' only</li> <li>5. Observe filtered product results</li> <li>6. Verify all displayed products show organic certification badge/indicator</li> <li>7. Click on a filtered product to view details</li> <li>8. Verify product details show organic certification information</li> <li>9. Clear filter and observe all products return</li> <li>10. Test edge case: Apply organic filter when no organic products are in current category</li> </ol>
<b>Expected Results</b>	<p>Filter correctly displays only certified organic products</p> <p>All filtered products display organic certification indicator</p> <p>Product details page shows certification status clearly</p> <p>Non-certified products are excluded from filtered results</p> <p>Filter can be easily applied and removed</p> <p>Empty results show appropriate message when no organic products exist in filter criteria</p>
<b>Acceptance Criteria</b>	<p>Organic certification status is accurately maintained for each product</p> <p>Filter works consistently across all product categories</p> <p>Visual indicators clearly distinguish certified vs non-certified products</p> <p>Filter combines logically with other filters (category, price, availability)</p> <p>Producers can set certification status when listing products</p>

**TC-015: As a customer, I want to see allergen warnings clearly displayed so that I can avoid products that may harm me or my family.**

<b>Test Case ID</b>	TC-015
<b>User Story</b>	As a customer, I want to see allergen warnings clearly displayed so that I can avoid products that may harm me or my family.
<b>Stakeholder</b>	Customer (All types)
<b>Priority</b>	Critical
<b>Description</b>	Validates that allergen information is prominently displayed for food safety and customer protection, supporting compliance with food safety regulations.
<b>Preconditions</b>	<p>Products have allergen information recorded</p> <p>At least 3 products contain common allergens (dairy, eggs, nuts, gluten)</p> <p>Some products have no allergens</p>
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Browse to a product containing dairy: Cheddar Cheese</li> <li>2. View product details page</li> <li>3. Locate allergen information display</li> </ol>

	4. Verify 'Contains: Milk' is clearly displayed 5. Browse to a product containing multiple allergens: Walnut Bread 6. Verify multiple allergens are listed: 'Contains: Wheat (Gluten), Nuts (Walnuts)' 7. Browse to a product with no allergens: Fresh Apples 8. Verify allergen section displays 'No common allergens' or equivalent 9. Search for products containing 'nuts' 10. Verify search results or filters allow allergen-based searching
<b>Expected Results</b>	Allergen information is displayed prominently on product details page Allergen warnings use clear, standard language Multiple allergens are listed individually Products without allergens clearly state this Allergen information is visible before adding to cart Visual design makes allergen warnings stand out (e.g., warning icon, distinct colour) All 14 major allergens recognised by UK law can be specified
<b>Acceptance Criteria</b>	Producers must provide allergen information when listing products Allergen information cannot be omitted for food products Display meets food safety labelling requirements Customers can filter products by allergen presence/absence System prevents ordering without viewing allergen information

**TC-016: As a producer, I want to set seasonal availability for my products so that customers know when items are in season.**

<b>Test Case ID</b>	TC-016
<b>User Story</b>	As a producer, I want to set seasonal availability for my products so that customers know when items are in season.
<b>Stakeholder</b>	Producer
<b>Priority</b>	High
<b>Description</b>	Validates that producers can manage seasonal availability without technical expertise and that customers see accurate seasonal indicators.
<b>Preconditions</b>	Producer is logged in Producer has multiple products with different seasonal patterns System supports seasonal availability settings
<b>Test Steps</b>	1. Navigate to product management 2. Select product: Strawberries 3. Edit product availability 4. Set availability to 'In Season' 5. Set seasonal dates: June - August 6. Save changes

	7. Select product: Stored Potatoes 8. Set availability to 'Available Year-Round' 9. Save changes 10. Log out and log in as customer 11. Browse products and observe seasonal indicators 12. Verify Strawberries show 'In Season' badge 13. Verify seasonal dates are displayed (e.g., 'Available: June - August') 14. Verify year-round products don't show seasonal restrictions
<b>Expected Results</b>	Producers can easily set seasonal availability Seasonal date ranges can be specified Products marked as seasonal display appropriate indicators to customers Out-of-season products are hidden or marked as unavailable In-season products are highlighted or badged System can automatically update availability based on current date and seasonal settings
<b>Acceptance Criteria</b>	Seasonal settings are intuitive for non-technical users Automatic date-based availability updates work correctly Customers cannot order out-of-season products Seasonal information educates customers about local food systems Producers receive reminders when seasonal products are about to become available

**TC-017: As a community group representative, I want to place bulk orders from multiple producers for catering needs.**

<b>Test Case ID</b>	TC-017
<b>User Story</b>	As a community group representative, I want to place bulk orders from multiple producers for catering needs.
<b>Stakeholder</b>	Customer (Community Group)
<b>Priority</b>	Medium
<b>Description</b>	Validates that community groups can create accounts and place larger orders with multiple suppliers for institutional catering purposes.
<b>Preconditions</b>	Community group account type exists in system Multiple products are available from different producers Bulk ordering functionality is enabled
<b>Test Steps</b>	1. Register as community group: St. Mary's School 2. Provide organization details and charity/education status 3. Complete registration with institutional email: catering@stmarys-school.org.uk 4. Log in as community group account



	<ul style="list-style-type: none"> <li>5. Browse and add multiple products with larger quantities: 50 kg potatoes, 30 litres milk, 20 kg carrots</li> <li>6. Products come from 3 different producers</li> <li>7. Proceed to checkout</li> <li>8. Enter delivery address: school address</li> <li>9. Select delivery date allowing for producer lead times</li> <li>10. Add special instructions: Delivery to kitchen entrance, contact kitchen manager</li> <li>11. Review order summary showing breakdown by producer</li> <li>12. Complete payment for bulk order</li> <li>13. Receive confirmation with producer contact details for coordination</li> </ul>
<b>Expected Results</b>	<p>Community group can register with appropriate account type</p> <p>System accepts larger quantity orders</p> <p>Checkout handles multi-vendor bulk orders</p> <p>Special delivery instructions can be provided</p> <p>Order confirmation includes all relevant supplier contacts</p> <p>Account may have different features (e.g., invoice payment, bulk discounts if applicable)</p> <p>Producers receive bulk order notifications with lead time</p>
<b>Acceptance Criteria</b>	<p>Community group accounts are distinguished from individual customers</p> <p>Bulk quantities are validated against producer capacity</p> <p>Special delivery requirements are communicated to all producers</p> <p>Payment terms may differ for institutional buyers</p> <p>System facilitates coordination between multiple suppliers and institution</p>

**TC-018: As a restaurant owner, I want to establish regular weekly orders so that I can simplify sourcing local ingredients.**

<b>Test Case ID</b>	TC-018
<b>User Story</b>	As a restaurant owner, I want to establish regular weekly orders so that I can simplify sourcing local ingredients.
<b>Stakeholder</b>	Customer (Independent Restaurant)
<b>Priority</b>	Medium
<b>Description</b>	Validates that business customers can set up recurring orders to reduce administrative overhead of managing multiple small supplier relationships.
<b>Preconditions</b>	<p>Restaurant business account is created and verified</p> <p>Multiple products from various producers are available</p> <p>Recurring order functionality exists</p>
<b>Test Steps</b>	1. Log in as restaurant account: The Clifton Kitchen

	<ol style="list-style-type: none"> <li>2. Create initial order with required weekly ingredients</li> <li>3. Add products from multiple producers: fresh vegetables, dairy, bakery items</li> <li>4. Set specific quantities for each item</li> <li>5. Before completing checkout, select 'Make this a recurring order'</li> <li>6. Set recurrence: Every Monday</li> <li>7. Set delivery date: Every Wednesday</li> <li>8. Review recurring order summary</li> <li>9. Confirm recurring order setup</li> <li>10. View 'Recurring Orders' management page</li> <li>11. Verify order appears with next scheduled date</li> <li>12. Modify next week's order (increase quantity of one item)</li> <li>13. Verify modification applies only to next order, not the template</li> </ol>
<b>Expected Results</b>	<p>Restaurant can create recurring order template</p> <p>Recurrence schedule (weekly, fortnightly) can be set</p> <p>Delivery day can be specified</p> <p>Recurring order automatically generates new orders on schedule</p> <p>Each scheduled order can be individually modified before confirmation</p> <p>Producers receive advance notice of recurring orders</p> <p>Restaurant can pause, modify, or cancel recurring orders</p> <p>System handles producer availability changes in recurring orders</p>
<b>Acceptance Criteria</b>	<p>Recurring order template maintains product selections and quantities</p> <p>Automatic order generation respects producer lead time requirements</p> <p>Individual order instances can be edited without affecting template</p> <p>Payment is processed for each recurring order instance</p> <p>Notifications are sent to restaurant before each order processes</p> <p>Unavailable products in recurring orders trigger alerts</p>

**TC-019: As a producer, I want to communicate surplus produce with discounts so that I can reduce food waste.**

<b>Test Case ID</b>	TC-019
<b>User Story</b>	As a producer, I want to communicate surplus produce with discounts so that I can reduce food waste.
<b>Stakeholder</b>	Producer
<b>Priority</b>	Medium
<b>Description</b>	Validates the surplus produce feature allowing producers to offer last-minute discounts on excess inventory to prevent food waste.
<b>Preconditions</b>	<p>Producer is logged in</p> <p>Producer has products with stock that needs to be sold quickly</p> <p>Surplus produce feature is enabled</p>
<b>Test Steps</b>	1. Navigate to product management

	<ol style="list-style-type: none"> <li>2. Select product with surplus stock: Lettuce (50 heads available, best before 3 days)</li> <li>3. Click 'Mark as Surplus' or 'Last Minute Deal'</li> <li>4. Set discount percentage: 30% off</li> <li>5. Set expiry date for deal: 48 hours</li> <li>6. Add note: 'Perfect condition, must sell quickly to avoid waste'</li> <li>7. Save surplus listing</li> <li>8. Verify product appears in 'Surplus Deals' section for customers</li> <li>9. Log in as customer</li> <li>10. Navigate to 'Surplus Deals' or 'Last Minute Offers'</li> <li>11. View discounted lettuce with clear discount badge</li> <li>12. Observe original and discounted price</li> <li>13. Add surplus item to cart and verify discounted price applies</li> <li>14. Complete purchase at reduced price</li> </ol>
<b>Expected Results</b>	<p>Producer can mark products as surplus with discount</p> <p>Surplus products appear in dedicated section for customers</p> <p>Discount is prominently displayed</p> <p>Urgency is communicated (time remaining, best before date)</p> <p>Discounted price is correctly calculated and applied at checkout</p> <p>Surplus deals are highlighted in customer interface</p> <p>Deals expire automatically after specified time</p> <p>System supports community food waste reduction objectives</p>
<b>Acceptance Criteria</b>	<p>Discount percentage is validated (e.g., 10-50% range)</p> <p>Surplus items maintain all quality and allergen information</p> <p>Deal expiry is enforced automatically</p> <p>Producers can remove surplus status if stock sells out</p> <p>Customers receive notifications about surplus deals from favourite producers</p> <p>Analytics track food waste reduction impact</p>

**TC-020: As a producer, I want to share recipes and farm stories so that I can engage with the community and educate customers.**

<b>Test Case ID</b>	TC-020
<b>User Story</b>	As a producer, I want to share recipes and farm stories so that I can engage with the community and educate customers.
<b>Stakeholder</b>	Producer
<b>Priority</b>	Low
<b>Description</b>	Validates educational features allowing producers to share seasonal recipes, storage guidance, and farm stories to strengthen community connections.
<b>Preconditions</b>	<p>Producer is logged in</p> <p>Content management feature is available for producers</p> <p>Products exist that can be linked to recipes</p>

<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to 'Content' or 'Farm Stories' section</li> <li>2. Click 'Add New Recipe'</li> <li>3. Enter recipe title: Roasted Root Vegetable Medley</li> <li>4. Add recipe description and ingredients list</li> <li>5. Link recipe to products: Carrots, Parsnips, Potatoes (from producer's inventory)</li> <li>6. Upload recipe image</li> <li>7. Add cooking instructions</li> <li>8. Select seasonal tag: Autumn/Winter</li> <li>9. Publish recipe</li> <li>10. Create farm story post about harvest season</li> <li>11. Add photos from farm</li> <li>12. Publish story</li> <li>13. Log in as customer</li> <li>14. View product page for Carrots</li> <li>15. Observe 'Recipe Suggestions' section showing linked recipes</li> <li>16. Click through to view full recipe</li> <li>17. Navigate to producer profile page</li> <li>18. View farm stories and educational content</li> </ol>
<b>Expected Results</b>	<p>           Producer can create and publish recipes            Recipes can be linked to specific products            Recipes appear on relevant product pages            Farm stories are visible on producer profile            Content includes images and formatted text            Seasonal tags help organise content            Customers can access educational content easily            Content strengthens producer-customer connection         </p>
<b>Acceptance Criteria</b>	<p>           Recipe format is user-friendly and readable            Product links are clickable and lead to purchase options            Content is appropriately moderated            Customers can save favourite recipes            Content supports local food education objectives            Storage guidance helps customers maximise product freshness         </p>

**TC-021: As a customer, I want to view my order history so that I can reorder favourite products and track past purchases.**

<b>Test Case ID</b>	TC-021
<b>User Story</b>	As a customer, I want to view my order history so that I can reorder favourite products and track past purchases.
<b>Stakeholder</b>	Customer (All types)
<b>Priority</b>	High
<b>Description</b>	Validates that customers can access complete order history with details and have the ability to quickly reorder previous purchases.

<b>Preconditions</b>	Customer is logged in Customer has completed at least 3 orders in the past Orders include various products and producers
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to 'My Account' or 'Order History'</li> <li>2. View list of past orders sorted by date (most recent first)</li> <li>3. Observe each order displays: order number, order date, delivery date, producer names, total amount, order status</li> <li>4. Click on a completed order to view full details</li> <li>5. View itemised list of products with quantities and prices</li> <li>6. View delivery address and payment information (partially masked)</li> <li>7. Click 'Reorder' button on a previous order</li> <li>8. Observe items are added to current cart</li> <li>9. Verify product availability is checked</li> <li>10. Adjust quantities if needed</li> <li>11. Proceed to checkout</li> </ol>
<b>Expected Results</b>	Order history is accessible and complete Orders are sorted chronologically Full order details are retrievable Reorder function simplifies repeat purchases Unavailable products in reorder are flagged Customer can filter orders by date range or producer Order status is clearly indicated Multi-vendor orders show producer breakdown
<b>Acceptance Criteria</b>	All historical orders are permanently accessible Order details match original purchase information Reorder function handles product availability changes gracefully Payment information is secure and appropriately masked Order history supports customer record-keeping needs Can download order receipts for past purchases

**TC-022: As a system administrator, I want to ensure secure authentication so that user accounts and data are protected.**

<b>Test Case ID</b>	TC-022
<b>User Story</b>	As a system administrator, I want to ensure secure authentication so that user accounts and data are protected.
<b>Stakeholder</b>	System (Security Requirement)
<b>Priority</b>	Critical
<b>Description</b>	Validates authentication and authorisation mechanisms ensuring secure access control for all user types with appropriate permissions.
<b>Preconditions</b>	System is configured with authentication system User roles exist: Customer, Producer, Community Group, Restaurant, Admin

	Test accounts exist for each role
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Test Case 1: Password Security</li> <li>2. Attempt to register with weak password: '123'</li> <li>3. Verify system rejects and shows password requirements</li> <li>4. Register with strong password meeting requirements</li> <li>5. Verify password is hashed in database (not stored as plain text)</li> <li>6. Test Case 2: Login Security</li> <li>7. Attempt login with incorrect password</li> <li>8. Verify appropriate error message (without revealing if user exists)</li> <li>9. Attempt login with correct credentials</li> <li>10. Verify successful authentication and session creation</li> <li>11. Test Case 3: Authorisation</li> <li>12. Log in as customer</li> <li>13. Attempt to access producer-only features (add/edit products)</li> <li>14. Verify access is denied with appropriate error</li> <li>15. Log in as producer</li> <li>16. Verify access to producer features is granted</li> <li>17. Attempt to view other producer's order details</li> <li>18. Verify access is denied</li> <li>19. Test Case 4: Session Management</li> <li>20. Log in and establish session</li> <li>21. Close browser and reopen</li> <li>22. Verify session persists if 'remember me' was selected</li> <li>23. Log out explicitly</li> <li>24. Verify session is terminated and protected pages require re-login</li> </ol>
<b>Expected Results</b>	<p>Password policy is enforced (minimum length, complexity)</p> <p>Passwords are securely hashed using industry-standard algorithm</p> <p>Failed login attempts are logged</p> <p>Brute force protection is in place (rate limiting)</p> <p>Authorisation checks prevent unauthorised feature access</p> <p>Users can only access their own data</p> <p>Sessions are managed securely with appropriate timeouts</p> <p>Logout fully terminates session</p>
<b>Acceptance Criteria</b>	<p>Role-based access control (RBAC) is appropriately implemented</p> <p>Passwords meet minimum security standards</p> <p>SQL injection attempts are prevented</p> <p>Cross-site scripting (XSS) is mitigated</p> <p>Session tokens are secure and unguessable</p> <p>Security logging captures authentication events</p>

**TC-023: As a producer, I want to receive a notification when stock for a product runs low so that I can restock before orders fail.**

<b>Test Case ID</b>	TC-023
<b>User Story</b>	As a producer, I want to receive a notification when stock for a product runs low so that I can restock before orders fail.
<b>Stakeholder</b>	Producer
<b>Priority</b>	Medium
<b>Description</b>	Validates inventory management alerts that notify producers when product stock levels reach defined thresholds.
<b>Preconditions</b>	Producer is logged in Products have stock quantities tracked Low stock threshold feature is enabled Notification system is configured
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to product management</li> <li>2. Edit product: Fresh Eggs</li> <li>3. Set current stock: 50 dozen</li> <li>4. Set low stock threshold: 10 dozen</li> <li>5. Save settings</li> <li>6. Simulate orders that reduce stock to 12 dozen (above threshold)</li> <li>7. Verify no low stock alert</li> <li>8. Simulate additional orders reducing stock to 9 dozen (below threshold)</li> <li>9. Verify low stock notification is generated</li> <li>10. Check notification centre/dashboard for alert</li> <li>11. Observe alert message: 'Low Stock Alert: Fresh Eggs - Only 9 dozen remaining'</li> <li>12. Update stock to 40 dozen</li> <li>13. Verify low stock alert is cleared/resolved</li> </ol>
<b>Expected Results</b>	Producer can set low stock thresholds per product System monitors stock levels automatically Notification is generated when stock falls below threshold Alert is displayed in producer dashboard Email notification is sent to producer (optional setting) Alert includes product name and current stock level Alert is resolved when stock is replenished above threshold Prevents accepting orders for out-of-stock items
<b>Acceptance Criteria</b>	Stock tracking is accurate and real-time Threshold settings are flexible per product Multiple notification methods are supported Alerts help prevent order fulfilment failures System can temporarily hide products from customer view when out of stock Stock levels are automatically decremented when orders are placed

**TC-024: As a customer, I want to rate and review products so that I can share my experience and help other customers make informed decisions.**

<b>Test Case ID</b>	TC-024
<b>User Story</b>	As a customer, I want to rate and review products so that I can share my experience and help other customers make informed decisions.
<b>Stakeholder</b>	Customer (All types)
<b>Priority</b>	Medium
<b>Description</b>	Validates the review and rating system allowing customers to provide feedback on purchased products, supporting community trust and quality assurance.
<b>Preconditions</b>	Customer is logged in Customer has completed at least one order that has been delivered Review system is enabled At least one delivered order contains product: Organic Tomatoes
<b>Test Steps</b>	<ol style="list-style-type: none"><li>1. Navigate to order history</li><li>2. Open a completed/delivered order</li><li>3. Locate product: Organic Tomatoes</li><li>4. Click 'Write Review' or 'Rate Product'</li><li>5. Enter star rating: 5 stars</li><li>6. Enter review title: Excellent quality and flavour</li><li>7. Enter review text: These tomatoes were incredibly fresh and flavourful. Perfect for our family's salads. Will definitely order again.</li><li>8. Submit review</li><li>9. Navigate to product page for Organic Tomatoes</li><li>10. Verify review appears in product reviews section</li><li>11. Observe average rating is updated to reflect new review</li><li>12. Verify review shows customer name (or anonymous option), date, and verified purchase badge</li><li>13. Test edge case: Attempt to review product from order not yet delivered</li><li>14. Verify system prevents reviewing until order is delivered</li><li>15. Test edge case: Attempt to review same product multiple times</li><li>16. Verify system allows only one review per product per customer</li></ol>
<b>Expected Results</b>	Customer can submit reviews only for purchased products Reviews are linked to delivered orders (verified purchase) Rating (1-5 stars) and text review can be submitted Reviews appear on product page for other customers to view Average rating is calculated and displayed Customer name is displayed (or anonymous option is honoured) Review date is shown System prevents duplicate reviews from same customer Reviews cannot be submitted until order is delivered
<b>Acceptance Criteria</b>	Review system verifies purchase before allowing submission Rating scale is clear and consistent (1-5 stars)



	Reviews are moderated for inappropriate content if needed Producers can respond to reviews Reviews contribute to product quality reputation Aggregate ratings help customers make informed choices System prevents fraudulent or spam reviews
--	---

**TC-025: As a system administrator, I want to monitor the network commission calculations so that I can ensure financial accuracy and generate reports.**

<b>Test Case ID</b>	TC-025
<b>User Story</b>	As a system administrator, I want to monitor the network commission calculations so that I can ensure financial accuracy and generate reports.
<b>Stakeholder</b>	System Administrator (Network Management)
<b>Priority</b>	High
<b>Description</b>	Validates that the 5% network commission is accurately calculated, recorded, and reportable across all transactions for network sustainability and financial compliance.
<b>Preconditions</b>	Admin user is logged in with appropriate permissions Multiple orders have been completed with various total values Financial reporting module is accessible Date range includes at least 2 weeks of orders
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. Log in as administrator</li> <li>2. Navigate to 'Financial Reports' or 'Network Commission' section</li> <li>3. Select date range: Previous 2 weeks</li> <li>4. Generate commission report</li> <li>5. View report showing: total order value for period, 5% commission amount, breakdown by order, producer payment amounts (95% per order), number of orders processed</li> <li>6. Select specific order from report</li> <li>7. View detailed breakdown: order total, commission calculation (5% of order total), producer payment per supplier in multi-vendor order, payment status</li> <li>8. Test calculation verification: If order total is £100, verify commission is £5.00 and producer payment is £95.00</li> <li>9. For multi-vendor order totalling £150 with 2 producers (£80 and £70 worth of items): verify total commission is £7.50 (5% of £150), producer 1 payment is £76.00 (95% of £80), producer 2 payment is £66.50 (95% of £70)</li> <li>10. Download report in CSV format for accounting software</li> <li>11. Generate monthly summary report</li> <li>12. View year-to-date commission totals</li> </ol>
<b>Expected Results</b>	Admin can access comprehensive financial reports 5% commission is correctly calculated on all orders

	<p>Commission calculations are accurate to 2 decimal places</p> <p>Multi-vendor order commissions are calculated on total order value</p> <p>Producer payments (95%) are correctly calculated per supplier</p> <p>Reports can be filtered by date range, producer, or order status</p> <p>Reports are exportable in multiple formats (PDF, CSV, Excel)</p> <p>All financial data is auditable and traceable to individual orders</p> <p>Running totals and period summaries are accurate</p> <p>Commission amounts match actual payment distributions</p>
<b>Acceptance Criteria</b>	<p>Financial calculations comply with accounting standards</p> <p>Commission split (5% network, 95% producer) is consistently applied</p> <p>Multi-vendor order payment distribution is correct</p> <p>Reports are suitable for tax compliance and business accounting</p> <p>Audit trail links all commission calculations to source orders</p> <p>System prevents unauthorised access to financial data</p> <p>Commission reporting supports network sustainability planning</p> <p>Rounding is handled consistently across all calculations</p>